

EXHIBIT C

Process

Hierarchical numeric ID: 1

Coded name:

Name: ARMS Context (CSD)

Comment: @Definition: Automated Rental Management System is the electronic data interchange (EDI) application and communications bridge between the Enterprise Rent-A-Car rental applications (ECARS-Enterprise Computer-Assisted Rental System and Claims Connection) and the application systems used by Rental Management Trading Partners (mostly insurance companies using ARMS/400, a VAN - Value-Added Network, or their own in-house application) and Direct Billing Trading Partners.

@Purpose: To accept EDI transmission envelopes from the trading partners, unpackage the transmissions, perform transaction data set edits, route to the appropriate distributed rental location's rental application system host computer platform and update the rental application database with the EDI transaction's information. To also, generate transaction data sets from the rental applications for sending to the trading partners.

@Notes: ARMS has a centralized data store for progressive transaction daily work files, received and sent transmission logs, error logs, billing logs, rental transaction database of information passed to date, trading partner transaction/rental transaction cross-reference, and trading partner profile files. ARMS has a distributed data store for transaction daily work files, error logs, trading partner transaction/rental transaction cross-reference and a subset of the duplicated images of the centralized trading partner profile files.

Also, the current transaction formats processed with some trading partners and within the majority of the ARMS software are using documented proprietary EDI transaction data set formats and transmission envelopes. However, some trading partners are exchanging information using ANSI (American National Standards Institute) X12 EDI standards for the 272 Rental Management transaction format that require translation to and from these proprietary EDI transaction data set formats, mapping, repackaging/unpackaging, and communication using an installed third-party software package, Any EDI integration software's "EDI Integrator".

Process

Hierarchical numeric ID: 1.1

Coded name:

Name: BA Automated Rental Management System

Comment: @Definition: This is a rental business area application system software set of processes that allows Automated Rental Management, rental reservations, rental location and rate maintenance notifications, invoice billing and electronic funds transfer (EFT) remittance advice transactions via EDI (Electronic Data Interchange) communications to be routed between trading partners and rental application and financial software systems within Enterprise Rent-A-Car.

Process

Hierarchical numeric ID: 1.1.1

Coded name:

ARMS Process Report

Name: AA Trading Partner Business Transactions

Comment: @Definition: This application area represents all EDI transmissions that are sent from the EDI Rental Management Trading partner users to be routed to the distributed application system for:

creating rental reservations; perform rental payment authorization maintenance for a current rental contract or reservation; or, messages to the rental branch office staff. Also, will process received electronic remittance advice detail in response to electronic billing invoice for a closed rental contract that they had authorized payment to be routed to the Cash Receipts financial application system.

Process

Hierarchical numeric ID: 1.1.1.1

Coded name:

Name: BT Request for Authorization Management by Customer

Comment: @Definition: The set of automated activities that are used in the creation of reservations and maintenance of the rental payment/direct billing authorization from an EDI transmission received from the Rental Management Trading Partner.

@Purpose: To allow the Rental Management Trading Partner an electronic means of creating rental reservations and managing their rental payment/direct billing responsibility authorization information for a vehicle rental.

Process

Hierarchical numeric ID: 1.1.1.1.1

Coded name:

Name: AUT Receive Transmission

Comment: @Definition: This represents any of the external (not between the multiple Enterprise Rent-A-Car AS/400 computer systems) connection-specific communication receiver programs that receive communication transactions from an electronic data interchange (EDI) trading partner or their VAN - Value-Added Network Service provider.

@Operational Method:

This program requests a program start from company, then continuously receives data from company until a record with *DOWN is received indicating that this was the last transmission before the trading partner's sender program shutdown. For every non-shutdown transmission received, a data queue entry is sent to DQAM70V1 so that the received transmission is placed into the Received Transmission Log file (AMTRNLOG) as an audit trail and then the transmission is then output to the connection-specific received transmissions for either production or test purposes work file (AMxxxxxn, where 'xxxxx' is the PROFILE ID of the trading partner or his value-added network for production or testing, and 'n' is the version number - currently all are the value '1').

@Notes: Each direct connect trading partner or VAN (Value-Added Network), has a PRODUCTION PROFILE ID and a TEST PROFILE ID.

Process

Hierarchical numeric ID: 1.1.1.1.1.1

Coded name: AMzzzRCV (zzz = Root of Van ID)

Name: PGM Receive Proprietary Transmission (AMzzzRCV)

ARMS Process Report

Comment: @Purpose: To receive proprietary EDI transmissions from any ARMS-connected VAN (Value-Added Network).

@Operational Method:

This program is evoked by a VAN's sender and then continuously awaits and receives data from the VAN until a record with *DOWN is received.

The input is logged and put in the appropriate receiver file.

For any program exception error, the *PSSR subroutine is executed to DUMP the current program and if the return code is equal to 8196 and the current time is outside of the 11:00pm - 3:00am time interval, call AMPSSR program to send a pager message to the ARMS primary and backup On-Call pagers so that manual intervention may be initiated.

@Notes: For specific ICF operations, see the record formats in the ARMICF file that this program uses for input.

The "real-time" receiver job abnormally ends execution when the VAN's shutdowns their communication environment at 1:00am nightly, 12:00am Sunday. This program would normally end when receiving a *DOWN request after having sent the VAN a *STOP request command from the AMSTOP2R program executed nightly at 12:50am. Any abnormal halts or ending of the production communication receiver job during their scheduled operational time currently notifies the ARMS On-Call staff in order that they may manually intervene to notify any ARMS-connected VAN to attempt to restart the critical communication process job.

Process

Hierarchical numeric ID: 1.1.1.1.1.2

Coded name: CLL818

Name: PGM Set Environment to Receive Proprietary Batch Transmission (CLL818)

Comment: @Definition: This is the ARMS Trading Partner Company - to -ARMS receive ARMS-connected Value-Added Network batch communication driver program (written in CL/400 - Original Program Model).

@Purpose: This program's purpose is to set up the environment and call the necessary program to receive an electronic data interchange batch transmission envelope formatted in Enterprise Rent-A-Car's ARMS- proprietary EDI formats, over a dialup communication line, to IBM ARMS-connected Value-Added Network electronic communication network mailbox. This mailbox contains one or more transmissions intended to be received by ARMS, sent from the ARMS Trading Partner Company's Insurance Claims Management application system's associated batch sender communications programs on their IBM mainframe computer network system.

@Operational Method:

- This program is delay until 7:00am prior to any further processing.
- The next step is to vary on (activate) the ARMS-connected Value-Added Network communication line, EDI controller (includes all downline attached configuration elements) and the EDI integration software device if any of these communication elements are found to be not ready.
- Add EDI integration software library to the batch job's library list.
- Then for a production batch receive, call/execute program AM***RCB with parameter '***01' (ARMS Trading Partner Company production Profile ID) to Receive

ARMS Process Report

ARMS Trading Partner Company Receive Batch file via ARMS-connected Value-Added Network with EDI integration software. This will write all records to ARMS Trading Partner Company Receive file and ARMS Receive Log file and send all records' transmission key value to the any ARMS Trading Partner Company Receiver Data Queue.

- The next step is to copy the records from the AM***RCB ARMS Trading Partner Company Received Batch Transmissions file to an archive file, AM***RCBSV and then clear the AM***RCB file.

- Repeat the call /execute, copy and clear for testing using the parameter 'T***T' (ARMS Trading Partner Company test Profile ID).

@Notes: Runs on the centralized ARMS application host computer system once per day, Monday through Saturday, after ARMS Startup's Restore Unprocessed Transaction batch job (L814) has completed, regardless if holiday. Scheduled via ROBOT/Scheduler REACT mode.

This program was created by the ETD- Emerging Technologies Department's (formerly Advanced Technologies Department) EDI project team, but is currently maintained primarily and the responsibility of the ARMS Application Development Department. This program is used to receive a batch transmission envelope that might contain the following Enterprise ARMS proprietary EDI transaction sets: Authorization maintenance (AT-Add and AT-Change), ER-Error, and PM-Payment Advice.

@Files: (CRUD)

- AM***RCB (-R-D)
- AM***RCBSV (C---)

Process

Hierarchical numeric ID: 1.1.1.1.1.5

Coded name: AM***RCB

Name: PGM Receive ARMS-connected Value-Added Network Batch for specified ARMS Trading Partner Companies (AM***RCB)

Comment: @Purpose: This program's purpose is to receive an electronic data interchange batch transmission envelope formatted in Enterprise Rent-A-Car 's ARMS-proprietary EDI formats, over a dialup communication line, to IBM ARMS-connected Value-Added Network electronic communication network mailbox. This mailbox contains one or more transmissions intended to be received by ARMS, sent from the specified ARMS Trading Partner Companies Insurance Company's ARMS-interfacing application system's associated batch sender communications programs on their host computer network system.

@Operational Method: This program is used to receive a batch transmission envelope that might contain the following Enterprise ARMS proprietary EDI transaction sets: Authorization maintenance (AT-Add and AT-Change), ER-Error, and PM-Payment Advice. It will retrieve the EDI integration software Profile record for account 'ERAC' and user ID 'ERACARM' from the IEPROFL IBM EDI integration software User Profile File, load the externally-defined local data area with program execution entry parameter values, and call the IBM EDI integration software licensed program product EDI library's IEIFEXEC main program for IBM EDI integration software BASE. This program will then read the IEIFEXEC program's output AM***SNB file records, that were "pulled" from specified ARMS Trading Partner Companies' send to ARMS mailbox on the IBM ARMS-connected Value-Added Network electronic communication network. Any production transmissions will be written to the AM***011 specified ARMS Trading Partner Companies production received transmission output file and its key sent to the DQ***011 specified ARMS Trading Partner Companies' Received Transmission Input Data Queue to its specific AM0010V1 program for transmission envelope validation. Any

ARMS Process Report

test transmissions will be written to the AM***011 specified ARMS Trading Partner Companies production received transmission output file and its key sent to the DQT***T specified ARMS Trading Partner Companies Received Transmission Input Data Queue to its specific AM0010V1 program for transmission envelope validation. For every input transmission record format received, each record format is written to the files AMTRNLOG (transactions received audit trail) and to file AMxxxxxxn and sent to data queue DQxxxxxxn (company 'xxxxxx' and version 'n' specific) for either production ("***01") or test ("T***T") purposes. For any program exception error, the *PSSR subroutine is executed to DUMP the current program and call AMPSSR program to send a pager message to the ARMS primary and backup On-Call pagers so that manual intervention may be initiated.

@Frequency: Once per day, Monday through Saturday after ARMS Startup's Restore Unprocessed Transaction batch job (L814) has completed, regardless if holiday.

@Notes: This program is called twice within a batch job's driver CL program, CLL818 ETD: ARMS Trading Partner Company Receive Batch program that runs on the centralized ARMS application host computer system. The first occurrence for this program execution with entry parameter to cause the program to perform only receive transmissions for the "***01" ARMS Trading Partner Company production processing. The second program execution with entry parameter "T***T" to cause the program to perform only receive transmissions for the "T***T" ARMS Trading Partner Company test processing. See documentation for CLL818 to determine when processed.

This program should be processing only when the pair of any ARMS-connected VAN communication sender program is active and the centralized ARMS application host computer system and its ARMS subsystem is active. Usually should be in operation from 3:00am to 1:00am, Monday through Tuesday and on Sunday from 7:30am to 11:00pm.

This program was created by the ETD- Emerging Technologies Department's (formerly Advanced Technologies Department) EDI project team, but is currently maintained primarily and the responsibility of the ARMS Application Development Department. This program is used to receive a batch transmission envelope that might contain the following Enterprise ARMS proprietary EDI transaction sets: Authorization maintenance (AT-Add and AT-Change), ER-Error, and PM-Payment Advice.

Process

Hierarchical numeric ID: 1.1.1.1.1.6

Coded name: EDMRLUA

Name: PGM Receive LU 6.2 Transmission (EDMRLUA)

Comment: @Definition: This is the ARMS Receiver LU 6.2 communication program.

@Purpose: This program's purpose is to receive an electronic data interchange transmission envelopes formatted in CIECA drafts for ANSI X.12 standard EDI formats, over a leased communication line, with an ARMS Trading Partner Insurance Company's automobile claims application system's associated sender communications program on their host computer network system.

@Operational Method: 1. Capture job attributes and set up library list, debug options, logging, traces, etc. 2. Retrieve parameters from data area with the same name as the job = same as the device description. (This allows control of various options without requiring that the initiator of the APPC ALLOCATE send parameters to control them.) 3. Set up communications line trace and/or ICF trace as specified. If the current user does not have sufficient authority, proceed without traces. If any other trace error, proceed without traces. 4. Execute either the EDISYS/STRCTLSCRIB (API: Start ctl script to call ARMS-connected Value-Added Network)

ARMS Process Report

or the EDISYS/STRRCVTRNB (API: Start ctl script; rcv/unwrap/transmission) command using appropriate parameters.

@Frequency: Heavy utilization between 6:00am - 3:00pm (all times are Central time zone), Monday through Friday, especially at 6:00am receiving batch burst of scheduled invoice payment advice (PM) transaction set transmissions. Additional heavy utilization from 10:00pm - 10:10pm receiving batch burst of scheduled invoice payment remittance advice (RM) transaction set transmission batches. Moderate utilization between 7:00am - 9:00am and 3:00pm - 7:00am Monday through Friday. Light utilization on all other active times, including weekends and holidays.

@Notes: This program was created and is maintained primarily by the Emerging Technologies Department (ETD - formerly Advanced Technologies Department) EDI project team. This program is used to receive all real-time business-event transaction data transmission sets for production or test information interchange. This is running as a prestart job because this will enable the programs involved in this job to be pre-started so that the programs and their resources remain open and active since each trading partner evokes the communication session each time they send a transmission envelope.

@Process Frequency: This program is used within two never-ending program prestart batch jobs on the centralized ARMS application host computer system. One job is Lxxxxx1R (where xxxxx is the full ARMS Company Profile ID) that executes EDxxxRC1 (where xxxxx is the full ARMS Company "root" Customer ID) - ETD: EC/EDI: ARMS X.12 Trading Partner Receiver Communication for Production. That program is started by remote via an APPC MC_ALLOCATE. In order for APPC (Advanced Program-to-Program Call) to trigger it properly, there should be a communications entry in the any EDI integration software subsystem which associates the device or remote location to JOBID EXCMNJOBID and user EDIOWN. This program can be executed from either a prestart job command within the AM***COMP (where *** is the Profile ID root value) program or by an ARMS subsystem "prestart" job that executes ED***RC1 program, that calls/executes this EDMRLUA program with a single entry parameter for the ARMS Profile ID value to cause the program to perform "receive" production transaction processing. The other job, ED***RCT, executes this program with entry parameter 'T***X' will cause the program to perform only "receive" test transaction processing. The production receiver job starts execution at usually on Sunday at 8:00am and stays active 24 hours per day through Saturday at 1:00am in order to do RARMS mirrored systems role swap when this "real-time" receiver job is normally ended. Any exception errors or abnormal ending that occur during the execution of the production communication receiver job on the centralized ARMS application host computer system during their scheduled operational time currently causes the notification of the ETD and ARMS primary and backup On-Call staff pagers. This is done so that they may manually intervene to attempt to restart the critical communication process job.

@Data Areas: (CRUD)

- EX || Company Id (-R--) (in EDI integration software system library)
- EXDBG (-R--) (in any EDI integration software system library)

Process

Hierarchical Numeric Ed. : 1.1.1.1.1.7

Coded name: EDMRMQA

Name: PGM Receive Alternate Protocol Transmission (EDMRMQA)

Comment: @Definition: This is the ARMS Receiver for alternate protocol communication program.

@Purpose: This program's purpose is to receive X12 standard EDI formats, over a leased communication line, with ARMS Trading Partner Insurance Company's

ARMS Process Report

automobile claims application system's associated sender communications program on their host computer network system.

@Operational Method: 1. Capture job attributes and set up library list, debug options, logging, traces, etc. 2. Retrieve parameters from data area with the same name as the job = same as the device description. (This allows control of various options without requiring that the initiator of the APPC ALLOCATE send parameters to control them.) 3. Set up communications line trace and/or ICF trace as specified. If the current user does not have sufficient authority, proceed without traces. If any other trace error, proceed without traces. 4. Execute either the EDISYS/STRCTLSCRB (API: Start control script to call ARMS-connected Value-Added Network) or the EDISYS/STRCVTRNB (API: Start control script; receive/unwrap transmission) command using appropriate parameters.

@Notes: This program is maintained by the ETD and is used to receive all real-time transaction sets for production or test environment. As of July 6, 1998, it has not been put into regular scheduled daily use because a specified ARMS EEDI Trading Partner Company's application changes to exchange with ARMS has not yet been implemented.

Process

Hierarchical numeric ID: 1.1.1.1.1.8

Coded name: EDMRUWA

Name: PGM Unwrap Envelope (EDMRUWA)

Comment: @Purpose: To unwrap received X12 EDI Transmission Envelope.

@Operational Method:

- Receive and unwrap an electronic data interchange transmission envelope formatted in ANSI X12 standard EDI formats.

- For any program exception error, the ATDPSSR and AMPSSR programs are called to send a pager messages to the ETD and ARMS primary and backup On-Call pagers so that manual intervention may be initiated.

@Notes: This program is maintained by the ETD. This program is used as a never-ending program batch job on the centralized ARMS application host computer system.

Currently used to receive transmissions over a leased communication line, with ARMS Trading Partner Insurance Company's automobile claims application system's associated sender communications program on their mainframe computer network system, based in Boston, Massachusetts.

Process

Hierarchical numeric ID: 1.1.1.1.1.9

Coded name: EDLRMIA

Name: PGM Receive Mapper Interface (EDLRMIA)

Comment: @Purpose: To drive and control the translation of the X12 EDI formats in the incoming transmission to proprietary formats.

@Operational Method:

- Declare programs variables

ARMS Process Report

- Change the any EDI integration software communication script name to the passed input parameter field value for COMPANY ID
- Allocate the receive interface files
- Wait for EDI transaction envelope
- IF transactions have been received, execute the any EDI integration software's Start Translate Connections Command to search for Log of Connection records which are eligible for incoming translation.
- IF translation error encountered, Notify ETD and ARMS on-call staff and end job.
- ELSE, Call the ARMS EDI Receive Interface program to write application records.
- IF program is running in debug mode, Copy receive interface files to their save files.
- Clear the receive interface files

ELSE, (transactions have not been received)

- Deallocate receive interface files.
- Send shutdown data queue entry to the receive unwrap input data queue.

@Notes: This program was created and is maintained primarily by the Emerging Technologies Department (ETD - formerly Advanced Technologies Department) EDI project team. This uses the any EDI integration software's mapping/translation utility third-party software package.

This program is submitted as a called never-ending program batch job with the job name 'LxxxxxIRM' (where "xxxxx" is the Trading Partner Profile ID) and specifies using the any EDI integration software system library's EDISFTWJOB DP job description on the centralized ARMS application host computer system by the 'L***011R' receiver job with the following input parameters:

5 Alphameric COMPANY ID
1 Alphameric DEBUG FLAG (possible values: "Y" (Yes) or "N" (No))

(12/17/98)

This object which is used in the receive mapper job has been changed to delay until midnight at shutdown before sending a shutdown data queue entry to the receive unwrapper job. This change was necessary because transactions received between 11:45 p.m. and midnight would not get functional acknowledgements sent back to ARMS Trading Partner Insurance Company and would not show up on our daily reporting because the receive unwrapper job wasn't active until midnight.

Process

Hierarchical numeric ID: 1.1.1.1.1.10
Coded name: EDZRCUA
Name: PGM Get Mapped Data (EDZRCUA)
Comment: @Purpose: This program's purpose is to check if any ANSI X12 EDI data has been received. This program also checks for shutdown.
@Operational Method:

This program is a called program that accepts three (3) parameter fields: Company Profile ID (input-5 alpha), Shutdown Flag (output-1 alpha) and Transactions Received Flag (output-1 alpha).

When executed, initialize the SCRIPT ID with the passed COMPANY PROFILE ID value concatenated with "R" (Receive).

Default load the TRANSACTIONS RECEIVED FLAG to "N" (No).

ARMS Process Report

If the passed COMPANY PROFILE ID is a production profile (ends in "1"), then change the receive mapping data queue name to 'DQ' + the first three characters of the company ID + 'RM1'. Else, change the receive mapping data queue name to 'DQ' + the second through fourth characters of the company ID + 'RMT'. Then continue the following processing until data has been received and mapped or a shutdown data queue entry has been received.

Receive a data queue entry from the receive mapping data queue on infinite wait. Once a data queue entry is received, change the receive data queue operation from infinite wait to 15 seconds.

Attempt to retrieve records from the Any EDI integration software Connection Log logical file by SCRIPT ID with Unwrapped Receive Status. If there are records, change the output transactions flag to 'Y' (Yes).

If the data entry received is '*DOWN', then load the SHUTDOWN FLAG parameter with 'Y' (Yes) and end the program.

For any program exception error, the *PSSR subroutine is executed to DUMP the current program and call AMPSSR program to send a pager message to the ARMS primary and backup On-Call pagers so that manual intervention may be initiated. Also, write the transaction the program was processing to the ERRORS file.

@Files: (CRUD)

- EXLLCPPR (-R--) * Any EDI integration software Connection Log by
Script Name with Unwrapped Receive Status (in EXTSYSF Any EDI
Integration software EDI Integrator Files library.)
- ERRORS (C---) * Errors Log (User-Controlled Open)

@Notes: This program is maintained by ETD. This program is used within two never-ending program batch jobs on the centralized ARMS application host computer system. One job executes this program with entry parameters to cause the program to perform only "receive" transaction processing, the other job executes this program with entry parameters to cause the program to perform only "send" transaction processing. The receiver jobs usually start execution at Monday through Saturday at 5:20am and Sunday at 1:30pm. The "real-time" receiver jobs abnormally end execution when the EDI trading partners shutdown their communication environment at 11:00pm. These programs will normally end when receiving a *STOP request from the AMSTOP2R program executed nightly at 12:50am. Any abnormal halts or ending of these jobs during their scheduled operational time currently notifies the ARMS On-Call staff in order that they may manually intervene to attempt to restart the critical communication process job.

Process

Hierarchical numeric ID: 1.1.1.1.1.12

Coded name: AM***COM

Name: PGM Receive/Send Proprietary Transmission for ARMS Trading Partner Company (AM***COM)

Comment: @Definition: This is the ARMS-to-ARMS Trading Partner Company (and vice-versa) SNUF (SNA Upline Facility) (IBM) 3270 terminal emulation device API transaction-based (triggered) communication program.

@Purpose: To receive or send an proprietary EDI formats, over leased ARMS-connected Value-Added Network lines, with the ARMS Trading Partner Company

ARMS Process Report

Insurance Company's Insurance Claims Management application system's associated sender and receiver communications programs on their host computer system.

@Operational Method:

This program requests a program start from company, then continuously awaits and receives data from ARMS Trading Partner Company's computer system until a record with *DOWN is received.

When input is received, each record is written to the files AMTRNLOG (transactions received audit trail) and to file AMxxxxxn (Company Profile ID 'xxxxx' and Version 'n' specific) for either production or test purposes.

For any program exception error, the *PSSR subroutine is executed to DUMP the current program and call AMPSSR program to send a pager message to the ARMS primary and backup On-Call pagers so that manual intervention may be initiated.

@Files: (CRUD)

- ARMSPR1 (-R--)
- ARMSPR3 (-R--)
- AM***011 (C--) (specified as ANDUMPRD)
- AMT***TT (C--) (specified as ANDUMTST)
- AMSEND (-R--)
- AMTRNLOG (C--)
- AMSNDLOG (C--)

@Notes: This program is maintained by ETD.

Process

Hierarchical numeric ID: 1.1.1.1.1.13

Coded name: EDxxxRC1

Name: PGM Prestart Job by Company for LU6.2 (EDxxxRC1)

Comment: @Definition: These are any programs that cause the jobs to be "prestarted" to define the system resources used in the communication prior to being sent any communications transmissions by a trading partner's remote system. A prestart job is a batch job that starts running before a program on a remote system sends a program start request. Prestart jobs are different from other jobs because they use prestart job entries to determine which program, class, and storage pool to use when they are started. Within a prestart job entry, you must specify attributes that the subsystem uses to create and manage a pool of prestart jobs.

@Purpose: The purpose of this program is to receive EDI formatted transactions from Rental Management Trading Partner and write them to the inbound control and detail files.

@Notes:

- This program is activated by ARMS Trading Partner Insurance Company, using an ALLOCATE verb. The corresponding ACCEPT verb makes the connection through ARMS-connected Value-Added Network using LU62.

- This program will received a complete EDI formatted 272 transaction set as one continuous long record. This long record is parsed out into 80 byte records and written to the detail file INBDTA. One corresponding control record is written to INBCTL for each complete transaction set received.

ARMS Process Report

- A last batch number is retrieved from dtaara EDIA001 and updated. This batch number is the key for the detail and control files.
- A Comm Profile record is also updated with a batch sequence number.
- A Confirm Request is received, and this program replies with a Confirmed response.
- A Deallocate is received and this pgm writes the control file record and then returns to the ACCEPT verb and waits for another ALLOCATE from ARMS Trading Partner Insurance Company. This sequence is then repeated.
- This program is a PRE-START program and is started with the sbs. It remains active even when not connected to ARMS Trading Partner Insurance Company. It will set forever on the ACCEPT verb, waiting for ARMS Trading Partner Insurance Company to ALLOC.
- This program does NOT go to EOJ when it receives the DEALLOCATE from ARMS Trading Partner Insurance Company, but re-establishes the connection by repeating the Accept, Receive, Confirmed, and Deallocate sequence.
- EOJ is indicated when a STOP record is received or a CPI-C return code indicates an error. It then must be manually restarted by the STRPJ command (start pre-start job).

Process

Hierarchical numeric ID: 1.1.1.1.1.14

Coded name: AMZRMIA

Name: PGM Receive X12 Interface (AMZRMIA)

Comment: @Purpose: To process X12 transmission set(s) within the receive interfaces files by mapping the elements from the transmission set data record to the associated ARMS proprietary record formats.

@Operational Method:

- Process the Electronic Commerce Customer Preference file (EDPREF3).
- IF transaction type equals Rental Management for Vehicle Replacement Rentals (X12 272)
 - Determine application transaction type (AK, AT, CN, EX, or ER)
 - Map associated X.12 element to corresponding proprietary record format
 - Write data format records to transmission received file (AMTRNLOG)
- IF transaction type equals Rental Management Remittance Advice for Vehicle Replacement Rentals (X12 820)
 - Map associated X.12 element to corresponding proprietary record format
 - Write data format records to transmission received file (AMTRNLOG)
- IF transaction type equals Rental Management for Vehicle Replacement Rentals (X12 824)
 - Determine application transaction type (RM-rejected, ER)
 - Map associated X.12 element to corresponding proprietary record format

ARMS Process Report

- Write data format records to transmission received file

(AMTRNLOG)

@Notes:

This program was created and is primarily maintained by the ETD - Emerging Technologies Department's EDI team.

ETD is currently working with the following X12 transaction sets that affect rental:

272 Rental Management - 2 maps, 1 for ARMS Trading Partner Company and a CIECA one which will be used by ARMS Trading Partner Company

824 Consolidate Payment - 2 maps, 1 ANSI X12 for ARMS Trading Partner Insurance Company and a CIECA X12 one which also will be used by a different ARMS Trading Partner Insurance Company.

820 Payment - CIECA map used by ARMS Trading Partner Company (who is a non-ARMS customer)

811 Consolidated Invoice - CIECA map for consolidated invoicing that ARMS Trading Partner Insurance Company and TVA are using.

832 Pricing Catalogue - map for ARMS Trading Partner Insurance Company which we are using to exchange branch rate and availability information.

The implementation guide for this was given to us by ARMS Trading Partner Company and that is what we used to get this going.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as any Enterprise Fleet Services customer.

@Files: (CRUD)

R272HDR1	(-R--)
R272NMH1	(-R--)
R272DTL1	(-R--)
R272LQ1	(-R--)
R272COM1	(-R--)
R272ER1	(-R--)
R997AK1	(-R--)
AMEDI011	(C---)
AMTRNLOG	(C---)
AMPGMERR	(C---)
R820HDR1	(-R--)
R820DTL1	(-R--)
R820ADX1	(-R--)
R820NTE1	(-R--)
AMXREFL1	(-R--)
EDPREF3	(- R--)
AMAUTD	(-R--)
ARMSPR1	(-R--)
AM805P01	(CR--)
AM806P01	(CR--)
AMSNDLOG	(-R--)
EDPREF3L1	(-R--)
R824HDR1	(-R--)
R824DTL1	(-R--)

ARMS Process Report

R824TED1 (-R--)
R824NAM1 (-R--)
R824NTE1 (-R--)
AM804P02 (-R--)
AMXREF (-R--)
AMCLSTB2 (-R--)
AM801P02 (-R--)

@Embedded Data/Constants:

FILE(AMEDIO11) TOFILE(AM) Used for override to appropriate trading partner/VAN production/test receive file

@Improvement Opportunity

- Use program status data structure to retrieve program name.

Process

Hierarchical numeric ID: 1.1.1.1.1.16

Coded name: AM_001

Name: PGM Retrieve ARMS Trading Partner Company Password (AM_001)

Comment: @Purpose: To retrieve the connection password for a ARMS Trading Partner Company from the data area so as to restrict the authority to these data areas to a certain specified number of people.

@Operational Method:

Retrieve data area AMSFPWPR if this is a production request.

Retrieve data area AMSFPWTS if this is a test request.

Return the password.

Process

Hierarchical numeric ID: 1.1.1.1.1.18

Coded name:

Name: DTQ Input for Receive Unwrapper (DQxxxRU1)

Comment:

Process

Hierarchical numeric ID: 1.1.1.1.1.19

Coded name:

Name: DTQ Input for Receive Mapper Interface (DQxxxRM1)

Comment:

Process

Hierarchical numeric ID: 1.1.1.1.1.20

Coded name: EDMRPIA

Name: PGM Receive X12 Transmission Batch (EDMRPIA)

Comment: @Purpose: To receive and unwrap the X12 transmission from a direct billing trading partner.

@Operational Method:

- Receive the actual transmission into a temporary buffer

ARMS Process Report

- CALL EDZ001A to retrieve direct billing trading partner's communication connection information. If the return code is '1', this means that there is no transmission to process otherwise use the returned communication NETWORK, PORT and SCRIPT information to retrieve the transaction data and put it into the application interface files.

- CALL AMZRMIA to convert retrieved data in interface files to proprietary formats for further processing.

@Notes: Currently, this program is called by a scheduled job (LTVA011RUM) that runs at 2:00 pm to receive a batch of transmissions from Tennessee Valley Authority (TVA)

Process

Hierarchical numeric ID: 1.1.1.1.1.21
Coded name: EDZ001A
Name: PGM Retrieve Communication Parameters (EDZ001A)
Comment: @Purpose: To retrieve the direct billing trading partner's communication information.

@Operational Method:

- Read file EDPREF1 to retrieve the direct billing trading partner's PORT, NETWORK and SCRIPT information.

- If no record is found a return code of 1 indicating nothing can be processed.

Process

Hierarchical numeric ID: 1.1.1.1.1.2
Coded name:
Name: AUT Validate AND Unpackage (CommRcvr - AM20/AM21)
Comment: @Definition: The automatic process of separating from the received and validated EDI transmission envelope the individual transaction data sets so each base transaction is processed further individually.

Process

Hierarchical numeric ID: 1.1.1.1.2.2
Coded name: AM0010V1
Name: PGM Validate Transmission (AM0010V1)
Comment: @Definition: The ARMS never-ending batch program to validate the completeness of a transmission envelope's contents.

@Operational Method:

At startup do the following steps:

1. Read, increment by 1 and update the DASDV1 data area value to indicate that another AM0010V1 NEP job has started.

2. Construct the derived input data queue name from "DQ" concatenated with the TRADING PARTNER/VAN PROFILE ID input parameter value concatenate with the constant "1".

3. Override the derived ARMS Input file (AMINPUT) to the ARMS Input TRADING PARTNER/VAN RECEIVER INPUT FILE Version Number 1. Then open the input file.

4. Override the derived ARMS Hold file (AMHOLD) to the ARMS Hold TRADING PARTNER/VAN RECEIVER INPUT FILE Version Number 1. Then open the hold file.

After startup, continue to receive the next data queue entry from the derived input data queue until program shutdown.

For a shutdown data queue entry received:

1. Read the DASDV1 data area value, decrement by 1, update it.
2. Send a shutdown data queue entry to the Unpackaging Input Data Queue (DQAMTRN1).
3. End the program/job.

For each non-shutdown data queue entry received:

1. Use its value to read by key the next TRADING PARTNER/VAN RECEIVER INPUT FILE record.
2. IF not end-of-file, check the retrieved record's FORMAT ID:
 - A. IF the record's FORMAT ID is part of the packaging (Transmission, Group, or Set Start/End), then load the record to that appropriate proprietary record format external data structure.
 - i. IF the SETE01 format has been read, check the derived TOTAL DETAIL FORMATS READ COUNTER is less than 1.
 - a. IF true, reject the transmission envelope
 - B. IF the record's FORMAT ID is NOT part of the packaging (i.e. the transaction's information), check if at least one Transmission, Group and Set Start record format was read.
 - i. IF false, then reject transmission with appropriate error code.
 - ii. IF true, then write it to the ARMS Hold TRADING PARTNER/VAN RECEIVER INPUT FILE Version Number 1 and increment by one the derived TOTAL DETAIL FORMATS READ COUNTER.
 - Validate if received transmission envelope is complete. If not valid, generate an Error Transaction (ER) data set record formats to the Transactions to be packaged for Sending (AMPACK) file and its key to the associated input (DQAMPKG) data queue, and write an associated record in the AMERRLOG file.
 - Generate a return Communication Acknowledgement (AK) transaction set for the valid transmission envelope received.

@Notes:

This program is only executed as an ARMS never-ending batch program in the ARMS subsystem only on the ARMS application centralized host computer system platform. This program is currently submitted by the ARMS Start-Up Job (CLL810) when executed on the system known as "RARMS" host computer system platform. This never-ending batch program ends normally when a shutdown data queue entry is received as input. Normally, this can indicate that its corresponding receiver or receiver's mapper-interface never-ending program batch jobs that had been previously active that day have processed a shutdown request.

It is submitted from the CLL810 ARMS Start-Up program with the following input parameter fields:

5 Alpha TRADING PARTNER/VAN PROFILE ID

ARMS Process Report

8 Alpha TRADING PARTNER/VAN RECEIVER INPUT FILE

@Files: (CRUD)

AMHOLD (CR-D) (HLxxxxxx1)

ARMSPR1 (-R--)

AMERRLOG (C---)

AMxxxxxx1 (-R--) (Where 'xxxxxx' is the connect-specific Trading Partner or VAN production or test Profile ID)

AMTRN (C---)

AMPACK (C---)

DASDV1 (-RU-) Data Area: number of AM0010V1 program/jobs running, used by AM0020V1 to check when last AM0010V1 program active has shutdown.

@Embedded Data/Constants:

'0123456789' for testing for valid numbers.

OVRDBF FILE(AMINPUT) TOFILE(AMCMPID1) to override to the appropriate AMxxxxxx1 file.

OVRDBF FILE(AMHOLD) TOFILE(HLCMPID1) to override to the appropriate HLxxxxxx1 file.

CLRPFM FILE(*LIBL/HLxxxxxx1) to execute a clear physical file member on the appropriate HLxxxxxx1 file.

Process

Hierarchical numeric ID: 1.1.1.1.2.3

Coded name: AM0020V1

Name: PGM Unpackage Transaction from Transmission (AM0020V1)

Comment: @Purpose: To unpackage received proprietary transmission and remove the transmission control formats (also referred to as transmission envelope) TSMS01, TSME01, SETS01, SETE01, GRPS01 and GRPE01.

@Operational Method:

1. Receive the next data queue entry for this program's input data queue (DQAMTRN1).

2. For every shutdown data queue entry, retrieve, decrement by 1, and update data area DASDV1 until new value is zero. When it is zero, send a shutdown data queue entry to the Edit Transaction Group/Set Record Formats data queue (DQAM25V1) and end the program. For further details about shutdown, refer to support activity "BAT Shutdown ARMS Environment".

3. For every non-shutdown data queue entry received:

A. Read all of the associated records (by the key value of the data queue entry) from the received packaged transmissions log file (AMTRN).

B. For each set start (SETS01) record format:

i. Attempt to use its PROFILE ID and its CUSTOMER TRANSACTION ID to retrieve an existing associated ARMS Cross-Reference File record. IF found, hold its VENDOR TRANSACTION ID and load its MACHINE ID, SOURCE ID, RENTAL LOCATION ID, RESERVATION ID, and TICKET ID values, along with the Transmission Group Start record format's GROUP TYPE CODE and the Transmission Group's Set Start record format's SET ACTION TYPE CODE values and this PROGRAM ID value and a ROUTING CODE of 'I', to their associated fields in the internal format APPD01.

ARMS Process Report

ii. Load the format, APPD01, and all associated subsequent transaction data set record formats' derived key value with the current record's key, concatenated with the Transmission Group Start record format's GROUP CONTROL ID and GROUP TYPE CODE, and the Transmission Group's Set Start record format's CUSTOMER TRANSACTION ID, and,

IF the ARMS Cross-Reference File record was retrieved successfully, its VENDOR TRANSACTION ID value.

C. For each record format read that is not a Transmission, Group, or Set Start/End record format,

If the group transaction type is not 'ER', write the record format with the derived key value to the output file, Transaction Sets to be Edited Daily Transaction file (AMSET)

else

If the group type in error as specified on the ERRD01 record format is not 'IN', write the record format with the derived key value to the output file, Received Errors Log file (AMRCVERR)

else

Write the record format with the derived key value to the output file, Received Billing Errors Log file (AMBILERR).

D. For each Transmission Group Set End (SETE01) record format, send the current transaction data set's derived concatenated key value to the Edit Transaction Group/Set Record Formats data queue (DQAM25V1) and finish reading the input file until the next Transmission Start (TSMS01), Group Start (GRPS01), or Set Start (SETS01) record format is read for the current read input key value.

@Notes:

This program is only executed as an ARMS never-ending batch program in the ARMS subsystem only on the ARMS application centralized host computer system platform. This program is currently submitted by the ARMS Start-Up Job (CLL810) with a single input parameter, VERSION, with the constant value of "1". This never-ending batch program ends normally when a set of shutdown data queue entries are received as input that indicate that each AM0010V1 never-ending program batch jobs that was previously active that day have processed a shutdown request.

@Files: (CRUD)

AMTRN	(-R--)
AMSET	(C---)
ARMSPR3	(-R--)
AMAUTD	(-R--)
AMRCVERR	(C---)
AMBILERR	(C---)

@Embedded Data/Constants:

'*DOWN00000000000000SD' is the key value used as output when sending a shutdown data queue entry to DQAM25V1.

'*DOWN' and 'SD' are the literal constants used to check the data queue entry value loaded into the ARMSKEY external data structure's data elements for COMPANY PROFILE ID and GROUP TYPE CODE to determine if a shutdown was received.

'*LIBL' is used as the literal constant to pass as an input parameter field value for the DATA QUEUE LIBRARY NAME for the calls to the 'QRCVDTAQ' and 'QSNDDTAQ' programs..

All of the possible ARMS Cross-Reference File Record's STATUS CODE values are used as literal constants in checking certain codes for conditional logic.

ARMS Process Report

@Improvement Opportunities:

'*DOWN00000000000000SD' could be used by many never-ending programs and the Send Data Queue Entry programs referenced from the Work with Data Queues menu. Therefore, place this value in an application system-wide data area and reference this external object.

Replace the literal constant '*LIBL' with this value in an application system-wide data area and reference this external object.

Process

Hierarchical numeric ID: 1.1.1.1.2.4

Coded name: AM0021V1

Name: PGM Unpackage Transaction from Transmission (AM0021V1-RM Only)

Comment: @Definition: This is a never-ending batch program to unpackage any remittance advice functional groups from a proprietary EDI transmission envelope into individual internal transaction data sets with one or more record formats.

@Purpose:

This program will edit and unpackage individual remittance detail transaction data sets within Remittance Advice ('RM') functional groups within received transmissions' proprietary EDI envelopes. This program also matches and edits each transaction set's Profile ID and Customer Transaction ID to an associated existing ARMS Trading Partner Profile File and an ARMS Rental Transaction Cross-Reference File record. This is used to validate if the retrieved files' records' attribute indication settings and the Cross-Reference Status Code value should allow such action, and outputs each validated individual internal transaction data set with an corresponding internal proprietary (APPD01) header record format and a unique associated record key value to ARMS Transaction Sets to be Validated (AMSET) file and its associated data queue (DQAM25V1). The output transaction set will be processed as input by the Validate Transaction Set Record Formats (AM0025V1) program

@Operational Method:

1. Receive the next data queue entry for this program's input data queue (DQAM21V1).

2. For every shutdown data queue entry, send a shutdown data queue entry to the Edit Transaction Group/Set Record Formats data queue (DQAM25V1) and end the program.

3. For every non-shutdown data queue entry received:

A. Using the data queue entry's Trading Partner Profile ID value, retrieve the associated ARMS Trading Partner Profile Application Specific Information File record.

B. Read each of the associated records (by their key value of the received data queue entry) from the ARMS Validated Received Proprietary EDI Transmission Envelopes to be Unpackaged file (AMTRN).

C. For each Transaction Set Start (SETS01) record format processed:
i. Load the AMTRN file record's key value (consisting of the TRADING PARTNER PROFILE ID and the TRANSMISSION CONTROL SEQUENCE ID NUMBER) to the AMSET file key and append to it the following:

a. the Functional Group Start (GRPS01) record format's GROUP CONTROL SEQUENCE NUMBER and the functional GROUP TYPE CODE.

b. the Transaction Set Start (SETS01) record format's CUSTOMER TRANSACTION ID and VENDOR TRANSACTION ID.

ii. Load the Internal Header Record Format (APPD01) fields with the appropriate data.

D. For each remittance detail (RMTD01) record format:

i. Validate its PAYMENT CODE. IF invalid, reject with ERROR

CODE='51'.

- Edit to ensure there is a non-blank EFT ID NUMBER on all paid remittances.

- Validate that any Rejected Payment Remittance Advice Detail Record Format does not have a value in the EFT ID NUMBER field.

ii. Use the retrieved AMTRN file record's key value's PROFILE ID and CUSTOMER TRANSACTION ID to attempt to retrieve an existing associated ARMS Cross-Reference File record. IF found, then validate that the Cross-Reference File record is the proper status. IF the read remittance detail's CUSTOMER TRANSACTION ID value does not match its retrieved ARMS Cross-Reference file record same field value, reject with ERROR CODE='82'. IF the status is valid, hold its VENDOR TRANSACTION ID and load its MACHINE ID, SOURCE ID, RENTAL LOCATION ID, RESERVATION ID, and TICKET ID values, along with the Transmission Group Start record format's GROUP TYPE CODE and the Transmission Group's Set Start record format's SET ACTION TYPE CODE values and this PROGRAM ID value and a ROUTING CODE of 'I', to their associated fields in the internal unpackaged transaction data set (APPD01) record format.

iii. Load the internal unpackaged transaction data set (APPD01) and all associated subsequent transaction data set record formats' derived key value with the current record's key, concatenated with the Transmission Group Start record format's GROUP CONTROL ID and GROUP TYPE CODE, and the Transmission Group's Set Start record format's CUSTOMER TRANSACTION ID, and,

IF the ARMS Cross-Reference File record was retrieved successfully, its VENDOR TRANSACTION ID value.

iv. Based on the trading partner profile, edit the remittance amount against the invoiced amount, and error the remittance detail record if the amount values are not equal. This comparison will not occur for RM reject or illegal transactions. The value of the profile attribute that indicates whether a trading partner sends remittances will be "X" for those customers that can short pay and so the totals need not match (e.g. TVA).

v. IF an error is detected, call error program AM0090V1, which will create an ERRD01 format.

vi. For each validated RMTD01 record format read, write the record format with the derived key value to the output file of Transaction Sets to be Edited Daily Transaction file (AMSET) and send its key as a data queue entry to DQAM25V1.

E. For a Remittance Advice Total record format (RMTT01) read, IF its TOTAL AMOUNT or the TOTAL TICKETS does not balance with all of the non-rejected Remittance Advice Detail record format records' PAYMENT AMOUNT sum or a derived record count, error the Remittance Total record format, but continue to process all the detail records which have not errored. Rejected Payment Transactions may exist within Paid Transactions batches. *Note, a remittance total format is not required when only one remittance advise is received and it's status is 'R'ejected.

F. For any record formats read other than RMTD01 and RMTT01 and the Transmission, Group and Set Start/End record formats, create an APPD01 format especially for it. Then write out that invalid format and save the AMSET key to write to DQAM25V1 when the SETE01 record format is read.

ARMS Process Report

@Notes:

This program is only executed as an ARMS never-ending batch program in the ARMS subsystem only on the ARMS application centralized host computer system platform. This program is currently submitted by the ARMS Start-Up Job (CLL810) with a single input parameter, VERSION, with the constant value of "1". This never-ending batch program ends normally when a set of shutdown data queue entries are received as input that indicate that each AM0010V1 never-ending program batch jobs that was previously active that day have processed a shutdown request.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer).

@Files: (CRUD)

AMTRN	(-R--)
AMSET	(C---)
AMIEBT	(-R--)
AMIEBH	(-R--)
ARMSPR1	(-R--)

@Embedded Data/Constants:

'0123456789' Named constant to check for numeric fields.

Valid values for BILLING INVOICE STATUS in the RMTD01 record format:

'P' = Payment (Can be in full or a partial payment).
'R' = Rejected (must have additional ERROR CODES provided)
'X' = Illegal (must have additional ERROR CODES provided)

All of the following are the possible ARMS Cross-Reference File Record's STATUS CODE values allowable are used as literal constants in checking certain codes for conditional logic to allow remittances to be further processed. However, 'P' is only valid when the ARMS Trading Partner Profile (ARMSPR1) flag is set accordingly.

'PABQ'
'B' = Rental Contract Electronically Billed at least once
'A' = Passed Electronic Billing Audit
'P' = Paid in full
'Q' = Rejected for Payment by Trading Partner

'*DOWN' and 'SD' (ShutDown) are the literal constants used to check the data queue entry value loaded into the ARMSKEY external data structure's data elements for COMPANY PROFILE ID and GROUP TYPE CODE to determine if a shutdown was received.

'AM0021V1' is used as a literal constant to pass as an input parameter field value to the calls of 'AMPSSR' and 'AM0090V1' programs.

'*LIBL' is used as the literal constant to pass as an input parameter field value for the DATA QUEUE LIBRARY NAME for the calls to the 'QRCVDTAQ' and 'QSNDDTAQ' programs..

@Improvement Opportunities:

ARMS Process Report

Replace the literal constant 'AM0021V1' with the Program Status Data Structure's data element for PROGRAM ID.

Replace the literal constant '*LIBL' with this value in an application system-wide data area and reference this external object.

Process

Hierarchical numeric ID: 1.1.1.1.2.5

Coded name: DQAM21V1

Name: DTQ Input for program AM0021V1 (DQAM21V1)

Comment: @Definition: DQAM21V1 is the data queue used to provide input to PGM AM0021V1 which processes remittance advises

Process

Hierarchical numeric ID: 1.1.1.1.2.6

Coded name: DQAMTRN1

Name: DTQ Input for program AM0020V1 (DQAMTRN1)

Comment: @Definition: DQAMTRN1 is the data queue used to provide input for PGM AM0020V1 which unpackages transactions.

Process

Hierarchical numeric ID: 1.1.1.1.2.7

Coded name:

Name: DTQ Input for Transmission Validation from Specific trading partner (DQxxxxx1)

Comment: @Definition: one of several company specific data queues used to provide input to a company specific AM0010V1 PGM used for transmission validation. Data Queues included are DQ***011, DQ***011, DQVN1011, DQVN2011, DQVN3011, and DQTVA011.

Process

Hierarchical numeric ID: 1.1.1.1.2.8

Coded name: DQAM70V1

Name: DTQ Input for Time Line Inquiry (DQAM70V1)

Comment: @Definition: DQAM70V1 is the data queue used to provide input to PGM AM0070V1 which provides a support information for oncall work.

Process

Hierarchical numeric ID: 1.1.1.1.2.12

Coded name: AMLCXLIT

Name: PGM Translate Location (AMLCXLIT)

Comment: @Purpose: To translate a current rental location to an old location or vice versa to deal with any past "Organization Split" or "Machine Split" conversion projects, such as the "Texas Group Split" of July 1995 or the "Bluegrass Group Split" of June 1997 or the "Group 53 split of October 1999". An example would be calling the program with a TRANSACTION DATE = 08/13/1998, LOCATION CODE = 'O' and CURRENT GROUP/BRANCH ID = '5211'. The returned TRANSLATED GROUP/BRANCH ID would be '6511'.

@Operational Method:

- This program is called with a single input/output AMLCXLIT external data structure with the following alphameric data elements and their usage:

ARMS Process Report

1	(Output)	Return Code
1	(Input)	OLD/NEW CURRENT VALUE
1	(Input)	CURRENT MACHINE ID
2	(Input)	CURRENT GROUP CODE
2	(Input)	CURRENT BRANCH CODE
4	(Input)	CURRENT GROUP/BRANCH
10	(Input)	CURRENT LOCATION
1	(Input)	CURRENT REGION
2	(Input)	CURRENT SUB-REGION
2	(Input)	CURRENT CITY
3	(Input)	CURRENT AREA
20	(Input)	VENDOR TRANSACTION ID
5	(Input)	ARMS PROFILE ID
20	(Input)	CUSTOMER TRANSACTION ID
8	(Input)	TRANSACTION DATE
10	(Output)	TRANSLATED LOCATION
4	(Output)	TRANSLATED GROUP/BRANCH
2	(Output)	TRANSLATED GROUP
2	(Output)	TRANSLATED BRANCH
8	(Output)	TRANSLATED MACHINE NAME
1	(Output)	TRANSLATED REGION
2	(Output)	TRANSLATED SUB-REGION
2	(Output)	TRANSLATED CITY
3	(Output)	TRANSLATED AREA

This pgm will accept a parameter list containing current location information and if a record is found in the ARMS Location Past Conversion Cross-Reference (AMLCXF) file, will return the translation information.

- IF the attempt to retrieve ARMS Rental Transaction Cross-Reference (AMXREF) file record fails, still attempt to perform the translate. Do this by running a test on the passed parameter's CURRENT TRANSACTION DATE.

-- IF all tests for transaction date checking fail, then move 'N' to RETURN CODE.

-- IF a test is successful, perform the routine to load the TRANSLATED MACHINE ID and LOCATION output parameter data elements.

@Notes: This program was written and is currently maintained by the ARMS Application Development Department. It is currently used by the ARMS Unpackage Transaction from Transmission Envelope (AM0020V1 and AM0021V1) programs, along with the ARMS Package Transmission Envelope (AM0120, AM0150, AM0151) programs along with the Financial Systems' AS Account schedules /JOB # 102 (EL0291) program. An interactive stub tester (TSAMLCXLT) program exists in production for debugging and testing purposes.

@Files: (CRUD)

AMLCXF (-R--) - Keyed by Old Location ID, along with its other logical access views,

Old Location	AMLCXFL0	- Keyed by
New Location	AMLCXFL1	- Keyed by
Old Group/Branch	AMLCXFL2	- Keyed by
New Group/Branch	AMLCXFL3	- Keyed by
Old Group and Old Branch	AMLCXFL4	- Keyed by

ARMS Process Report

New Group and New Branch	AMLCXFL5	- Keyed by
Old Group, Region and Sub-Region	AMLCXFL6	- Keyed by
New Group, Region and Sub-Region	AMLCXFL7	- Keyed by
Old Group and City	AMLCXFL8	- Keyed by
New Group and City	AMLCXFL9	- Keyed by
Old Group and Area	AMLCXFL10	- Keyed by
New Group and Area	AMLCXFL11	- Keyed by
Old Group and Region	AMLCXFL12	- Keyed by
New Group and Region	AMLCXFL13	- Keyed by

Process

Hierarchical numeric ID: 1.1.1.1.3

Coded name:

Name: AUT Validate Transaction (AM25 - AM46)

Comment: @Definition: The automatic process of transaction validation and database update on the centralized machine prior to dispatch to either the distributed machine or the Rental Management Trading Partner. This process is currently used for both the inbound and outbound processing of transactions.

Process

Hierarchical numeric ID: 1.1.1.1.3.1

Coded name: AM0025V1

Name: PGM Validate Transaction Set Record Formats (AM0025V1)

Comment: @Definition: This is the ARMS Transaction Set Record Format Edit never-ending batch ILE RPG program.

@Purpose:

The program edits a transmission set group type to ensure that the required formats are present and that no non-allowable record formats exist.

Edit errors from an Inbound (Received) transaction set will be logged and will cause error transaction sets to be generated and sent back to the associated sender.

Edit errors from an Outbound (to be Sent) or Transfer transaction set will be logged and the ARMS On-call support staff will be notified.

@Operational Method:

This program endlessly receives the non-keyed input data queue entries from the input data queue (DQAM25V1) that the either of the ARMS Unpackage Remittance/Non-Remittance Envelope programs (AM0021/AM0020V1) generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQAMSET1 data queue and end this program.

ARMS Process Report

For any non-shutdown data queue entry, check the non-shutdown key Group Type Code to be one of the following values:

- Validate Transaction Set Group Type
- Validate Allowable Record Format(s) within Transaction Set for a Group

Type

@Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters to execute only on the centralized ARMS host system platform ("RARMS"). Currently, all inbound (received), outbound (to be sent) and transfers are processed in this program as the proprietary EDI Transaction Sets group types.

IF any error is found, then do not process this transaction any further and if it is an inbound transaction set, then execute the ARMS Generate External Error (AM0090V1) program, ELSE, execute the ARMS Handle Internal Error (AM0097V1) program, passing the input parameters of this data queue entry's key data elements and the currently loaded ERROR CODE value. Do NOT output the transaction set's record formats to the output file nor send its data queue entry to the output data queue.

Called Programs Summary:

QRCVDTAQ --Read entry from DQAM62V1 data queue
QSNDDTAQ --Send entry to DQANDST data queue
AMPSSR --ARMS program error handler
AM0097V1 --Handle Internal Error
AM0090V1 --Generate External Error

@Files: (CRUD)

- AMSET (-R--)
- AMAUTD (-R--)

@Embedded Data/Constants:

'*LIBL' as the DATA QUEUE LIBRARY NAME value used in the execution of the 'QRCVDTAQ' and 'QSNDDTAQ' programs.

'DQAM25V1' and 'DQAMSET1' are the specified data queue names.

'AM0025V1' as the current PROGRAM ID.

'AM0090V1' and 'AM0097V1' are the executed programs, based on the type of error.

'SD' and '*DOWN' are the shutdown request GROUP TYPE CODE and COMPANY PROFILE ID values.

'OFF' and 'ON'

'I' (inbound), 'O' (outbound), and 'T' (transfer)

ERROR CODES: '16', '45', '69', '70', '71', and '90'.

GROUP TYPE CODES for execution of their associated routine:

ARMS Process Report

'AT', 'EX', 'PM', 'ER', 'AK', 'AC', 'IN', 'OF', 'RA', 'RC', 'RE',
'RN', 'CC', 'RM', 'CM', and 'VM'

Compile-Time arrays of:

..Valid transaction set GROUP TYPE CODES

..Valid RECORD FORMAT ID's for each transaction set's GROUP TYPE CODE

@Improvement Opportunities:

1.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') and ARMS Generate External Error ('AM0090V1') programs with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.

2.) Delete the constant 'AM0025V1' and replace its usage with the Program Status Data Structure data element for PROGRAM ID.

Process

Hierarchical numeric ID: 1.1.1.1.3.2

Coded name: AM0030V1

Name: PGM Validate Transaction Set Record Formats Data Types (AM0030V1)

Comment: @Purpose:

To edit a transaction set to ensure the all the required format data elements have values. Whether a data element is required or not is based on the trading partner profile.

@Operational Method:

- Wait for entry(s) to exist in DTAQ (DQAMSET1).
- When a shutdown request is received (group type - SD), log the occurrence, pass on the request to DTAQ (DQANDTL1) and wait for the second shutdown.
- When the second shutdown request is received (group type - SD), log the occurrence, pass the request to DTAQ (DQANDTL1) and end.
- When a non-shutdown request is received, read all associated data record(s) from the ARMS Detail Set Transaction File (AMSET) into the appropriate ARMS record format data structures.
- Validate a transaction set's group type record formats' data elements for profilable mandatory values existence.
- Validate data field values with the data format(s) against a domain of allowable values.
- IF any error is found, then do not process this transaction any further and if it is an inbound transaction set, then executed the ARMS Generate External Error (AM0090V1) program, ELSE, execute the ARMS Handle Internal Error (AM0097V1) program, passing the input parameters of this data queue entry's key data elements and the currently loaded ERROR CODE value. Do NOT output the transaction set's record formats to the output file nor send its data queue entry to the output data queue.

@Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters to execute only on the centralized ARMS host system

ARMS Process Report

platform ("RARMS"). Currently, all inbound (received), outbound (to be sent) and transfer transaction sets are processed in this program as proprietary EDI Transaction Set group types.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer); and AAA of California.

@Files: (CRUD)

```
-AMSET          (-R--)
-STATE          (-R--)
-AMCLSTBL       (-R--)
-CCSURCHG       (-R--)
-AMERRTBL       (-R--)
-AMXBCOL1       (-R--)
-AMICDTBL       (-R--)
-ARMMAP         (-R--)
-ARMSPR1        (-R--)
-ARMSPR4        (-R--)
-AMAUTD         (-R--)
```

@Embedded Data/Constants:

'0123456789' TO CHECK FOR VALID NUMERIC FIELDS VALUES

'0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ ' TO CHECK FOR VALID ALPHANUMERIC FIELDS VALUES

'*LIBL' as the DATA QUEUE LIBRARY NAME value used in the execution of the 'QRCVDTAQ' and 'QSNDDTAQ' programs.

'DQAMSET1' and 'DQANDTL1' are the specified data queue names.

'AM0030V1' as the current PROGRAM ID.

'AM0090V1' and 'AM0097V1' are the executed programs, based on the type of error.

'SD' and '*DOWN' are the shutdown request GROUP TYPE CODE and COMPANY PROFILE ID values.

'OFF' and 'ON '

'I'(nbound), 'O'(utbound), and 'T'(ransfer)

All allowable values for each data element

ERROR CODES: '01', '02', '03', '04', '05', '21', '68', '70', '78', and '82'

GROUP TYPE CODES for execution of their associated routine:

'AT', 'EX', 'CN', 'PM', 'ER', 'AK', 'AC', 'IN', 'OF', 'RA', 'RC', 'RE', 'RN', 'CC', 'RM', 'CM', and 'VM'

Compile-Time arrays of:

..Valid transaction set GROUP TYPE CODES

..Valid RECORD FORMAT ID's for each transaction set's GROUP TYPE CODE

ARMS Process Report

@Improvement Opportunities:

- 1.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') and ARMS Generate External Error ('AM0090V1') programs with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.
- 2.) Delete the constant 'AM0025V1' and replace its usage with the Program Status Data Structure data element for PROGRAM ID.
- 3.) Convert program from OPM RPG to ILE RPG.
- 4.) Replace the use of the AM2010V1 program with the AM1010V1 program called for "I"nquiry.

Process

Hierarchical numeric ID: 1.1.1.1.3.3

Coded name: AM0040V1

Name: PGM Validate Transaction Set Record Format Data (AM0040V1)

Comment: @Purpose: To perform the detail edits on any transaction set data element values against the centralized ARMS database and domain of valid values. Also, to assign a rental location to any new Rental Authorization Add transaction that is not authorizing an existing rental reservation or open ticket.

@Operational Method:

Wait for keys to exist in DTAQ (DQANDTL1), when key is received, one of three processes occur. Program terminates when group type (SD) shut down data queue entry is encountered and passed on for the second time to output data queue (DQAM46V1).

For a non-shutdown data queue entry, use its value to read all associated records' data values from the ARMS Detail Set Transaction File (AMSET) into the appropriate ARMS record format data structures.

Validate Transaction Set's record formats' data fields' contents for allowable values.

Validate Transaction Set's functional group type code and its record format's data field's contents against centralized ARMS Rental Transaction Cross-Reference Status and its associated database file fields to ensure that this transaction should be allowed to continue further downstream processing. Send message to ARMS On-Call and reject any Authorization Transfer transaction or outbound transaction to be sent that fails an edit.

IF any received inbound transmission with a specified Branch Claims Office value that does not exist in the Branch Claims Office Cross-Reference file, then assign the first three characters (root customer) of the trading partner's PROFILE ID to ENTERPRISE CUSTOMER ID and send a warning message to ARMS Technical Management (via SNDEMSG) of a transaction with a missing BCO X-Ref record so that they can determine if this BCO is to be added. Do NOT reject the transaction.

ELSE, load the internal routing record format (APPD01) with the retrieved internal ENTERPRISE CUSTOMER ID value.

IF any received inbound transmission with a invalid Date Of Birth value, then set the value to zeros and send a warning message "Invalid DOB" to ARMS Technical Management which is currently not acted upon by them.

ARMS Process Report

The messages sent to ARMS Technical along with the employee numbers to send the message to exist in file SNDMSG.

IF an authorization-add transaction set or a group-to-group authorized reservation transfer transaction set has a non-blank rental location (Group ID and Branch ID), then validate the rental location against the Daily Rental Office Locations (DROFF) file by the Rental Location ID. IF not found for the authorization-add/transfer, or, for an authorization-add, the location found indicates that it does prevent ARMS reservations, then use its non-blank Forward ARMS Reservations To Location. If the Forward ARMS Reservations To Location is blank or not found, then derive the closest available daily rental location and its platform.

IF a received unsolicited Authorization Add rental transaction set is received that is to authorize and create a currently non-existent rental reservation, derive and assign closest available rental location and its distributed application host platform for later routing to that platform.

To derive the closest available rental location we use RAS013A.

IF the authorization add/transfer transaction set is for Enterprise Fleet Management VIP rental customer, they will be able to send and generate ARMS authorized reservations to airport branches if that is the closest rental location. Any other trading partner authorization add/transfer cannot be routed to an airport location.

Derive that rental location's associated distributed application host platform by executing Derive Machine ID for Rental Location (AM2050V1) program for later routing.

IF a transaction's record formats are validated and processed successfully, then send its associated key value as output to the data queue (DQAM46V1) that is input to the next downstream process for ARMS Database Update and Routing (AM0046V1) program.

@Files: (CRUD)

AMSET	(CRU-)
ARMSPR1	(-R--)
AMAUTD	(-R--)
AMIEBT	(-R--)
AMXBCO	(-R--)
DROFLF1	(-R--)
DROFFL7	(-R--)
FILE80	(C---)

@Notes:

This program is submitted for execution as a never-ending batch program/job in the ARMS subsystem by the ARMS Startup Jobs program (CLL810) with no entry parameters to execute only on the centralized ARMS host system platform ("RARMS"). It will usually be active from 3:00am to 1:00am-Central. Currently, all inbound (received), outbound (to be sent) and transfer transaction sets are processed in this program as proprietary EDI Transaction Set group types.

IF any error is found, then do not process this transaction any further and if it is an inbound transaction set, then execute the ARMS Generate External Error (AM0090V1) program, ELSE, execute the ARMS Handle Internal Error (AM0097V1) program, passing the input parameters of this data queue entry's key data elements and the currently loaded ERROR CODE value. Do NOT output the transaction set's

ARMS Process Report

record formats to the file nor send its data queue to the output data queue.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer); and AAA of California.

@Embedded Data/Constants:

'7680' used as the default Claims Connection Rental Location.
'6A01' used for routing to this simulation default rental location within the production software environment.
'0123456789' used as valid numeric values in a CHECK operation.
'RTVMSGSTS' is the program executed to check if the e-Message System is available for sending messages.

'OVRPRTF FILE(QPPGMDMP) OUTQ(ARMSDUMP) OUTPTY(9) USRDTA(AM40PGMERR) FORMTYPE(ARMS) HOLD(*YES)' to cause generated RPG-formatted dump output to be retained for greater than a one day.

The following are 25-character ERROR MESSAGES subject header and detail line prefixed output to FILE80 for sending when executing MS0010 program:

'An invalid BCO was sent'
'Invalid BCO'
'Profile'
'transaction control ID'
'Invalid DOB was received'
'Invalid DOB'

@Improvement Opportunities:

1.) Remove as constant and store the default simulation rental location ID ('6A01') and the default production Claims Connection rental location ID in an external data area to externalize this location information for inquiry into the program.

2.) Should be able to replace the valid numeric values constant and its usage with an ILE built-in function or existing API.

3.) Should be able to duplicate the existing OS/400 Dump printer file QPPGMDMP to a different ARMSDUMP printer file that is formatted the same but has all of the specified keyword attributes that would allow the removal of the OVRPRTF OS/400 command execution.

Process

Hierarchical numeric ID: 1.1.1.1.3.4

Coded name: AM0046V1

Name: PGM Route Transaction / Update ARMS Centralized Database (AM0046V1)

Comment: @Purpose: To update the centralized ARMS database with transaction set data elements and forward the transaction set to the appropriate destination (trading partner, rental application, claims connection, or centralized financial system's cash receipts interface).

@Operational Method:

ARMS Process Report

- Maintain centralized ARMS application system database with input transaction set data elements.

- Route outbound transaction data set for external envelope packaging.

- Route inbound or transfer transaction data set for sending to distributed host platform.

- IF processing an RA Change, AT Change or an an Opening Rental Notification RN functional group type, then calculate the renter's age by subtracting their Date of Birth from the system date. IF the renter is underage for the state of the rental and the ticket is open and authorized, then execute the Generate Surcharge subroutine (GENSUR) to load the Approved Surcharge Detail (AMSURD) file. This subroutine calls AM1085V1 for underage surcharges and AM0096V1 for othe surcharges. An RA-Change for an ARMS Trading Partner Company will only be generated if the transaction has been confirmed. (See @Notes for more detailed information)

- On an open authorized ticket, if a different Rental Location ID is processed from an outbound or transfer transaction set, then execute the Add Government Surcharge (AM0096V1) program to load the Approved Surcharge Detail (AMSURD) file.

- IF any new approved surcharges were generated, then read this file for this rental transaction to generate a new automatic pseudo-rental Authorization Change (AT) transaction set for approved surcharges to be routed to a distributed rental application system host platform.

- IF a Customer Extension/Termination, functional group type EX is received from a Trading Partner Company for a defined Profile ID AND the BCO or Adjuster Code is different than on the ARMS centralized database, then generate a new automatic pseudo-rental Authorization Change (AT) transaction set containing only an adjuster detail (ADJD01) record format for an adjuster/BCO change to be routed to a distributed rental application system host platform. This is done to update Bill-To Customer information in the rental system database files for this rental transaction in the EC00ATV1 program processing, since the EC00EXV1 program processing does not update Bill-To Customer information in the rental system database files. (See @Improvement Opportunities number 3 below.)

- IF processing an Extension, move an "E" to the Action Code in the AM055V1 external data structure. When a termination date has been sent, update the Day Auth (AWAUDY) field in file AMAUTD.

Calculate the total number of days authorized by using the Termination Date and Rental Start Date adding 1 to the calculated days to receive the correct number of days authorized. If the calculated amount is greater than the Day Auth field, move an "E" to the Action Code in the AM055V1 external data structure.

If the calculated amount is less than or equal to the Day Auth field, move a ~~TT~~ to the Action Code in the AM055V1 external data structure.

- Based on the trading partner profile, generate new automatic rental Extension days transaction set in response to a Request for Extension (RE) to be routed to a distributed rental application system host platform.

- IF processing an Opening Rental Notification ('RN') functional group type, IF the existing customer number on the transaction set internal header is found in the ARMS BCO for RMS (Rental Management System) file write a record to the ARMS RMS (Rental Management System) File (AM002P) so that RMS will be notified of the open rental contract/ticket that they must manage from their centralized location.

IF processing a Rental End ('RN') transaction set AND this is the first closing rental notification (AMAUTD's Rental End Date = 0) AND IF the associated

ARMS Process Report

ARMSPR1 profile file record's Remittance Indication Flag (AMFG14) field is equal to 'A', then execute the ARMS Generate Transaction Credits (AM1080V1) program passing the Transmission Control Sequence ID Number, Group Control Sequence ID Number, Vendor Transaction ID and the Execution Code of 'R' (for Rental Notification).

- IF processing a Payment Advice ('PM') transaction set, then execute the ARMS Generate Transaction Credits (AM1080V1) program passing the Transmission Control Sequence ID Number, Group Control Sequence ID Number, Vendor Transaction ID and the Execution Code of 'P' (for Payment Advice).

-After the file updates and before the next data queue entry is read, and if the Group Type is not OF.

-Complete the build of the data queue entry record.

-From the APPD01 format, move the Profile Id, Customer Transaction Id, Vendor Transaction Id, -Group Type, Date, and Time to their appropriate fields in DQAM55V1 external data structure, AM055X.

-If Transaction Type (AM055X) is blank, move an I internal to the field.

-Send the data queue entry to DQAM55V1.

-Clear the AM055X format for the next record.

@Notes:

Currently, ALL transaction sets (received from trading partners, received from rental systems - i.e., inbound, outbound and transfers) will be processed by this program, after editing for updating to the database and routing (to a trading partner or a rental application system).

When the Rental Management Trading Partner sends an extension transaction set for an existing Vendor Transaction ID with a different Customer Transaction ID than the value in the ARMS Cross-Reference (AMXREF) file record's CUSTOMER TRANSACTION ID (XACTID) field, then this field change should occur and allow the AMXREF file record's XACTID field to be updated with the new/different Customer Transaction ID value. All incoming transaction sets should allow a change of the Customer Transaction ID. Remittance Advice ('RM') transaction sets will not be included because this would require a change in the way RM's are processed.

When a request for cancellation ('RC') transaction set is processed for an incompletely open rental contract/ticket (prewrite/unit-pend), this 'RC' transaction set is not sent to the trading partner. In most cases this is correct, however when the ARMS Cross-reference file record is made irretrievable by padding the rental location value with the repeating "X" character, the request for cancellation transaction set will be sent. For some profiled ARMS Trading Partner Companies, the RC is converted to a CN before being sent.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer); and AAA of California. Therefore, ARMS will receive a data queue entry to generate an electronic invoice (and create an associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) when the trading partner never sent to ERAC (via ARMS) any initial rental authorization.

When a Request for an Authorization is sent to ARMS Trading Partner Company after the initial request, and the transaction is still in an unconfirmed status (U), an RA-Add will be generated. IF the transaction is at a confirmed status (W) but still unauthorized an RA-Change will be generated even if no changes have

ARMS Process Report

been made to the data. the transaction is at a confirmed and Authorized status (R), an RA-Change will be generated only if there was a change in the data.

@Files: (CRUD)

- AMSET (-R--)
- ARMSPR1 (-R--)
- ARMSPR3 (-R--)
- AMXBCO (-R--)
- AMPGMCTL (-R--)
- AM002P (CR--) ARMS: RMS FILE
- AM003P01 (-R--) ARMS RMS BCO FILE
- AMAPPS (C--)
- AMPACK (C--)
- AMADJD (CRU-)
- AMAUTD (CRU-)
- AMCOMD (CR--)
- AMIEBD (CRU-)
- AMIEBH (CRU-)
- AMIEBT (CRU-)
- AMINSO (CRU-)
- AMRMTD (CRU-)
- AMRNTD (CRU-)
- AMRPRD (CRU-)
- AMRATD (CRUD)
- AMSURD (CR-D)

@Embedded Data/Constants:

'YES'
'NO '
'AM0046V1 '
'RECORD DELETED'
'XXXXXXXXXX' (to pad ARMS Rental Transaction Cross-Reference File
record's RENTAL LOCATION with X's to make it irretrievable)
'AUTHORIZATION EXPIRED' (as COMD01 record format's COMMENT comment text
Currently not referenced, unsure of intent.)

Enterprise Customer Numbers for ARMS Trading Partner Insurance Company-
Los Angeles Area Locations and ARMS Trading Partner Insurance Company's Branch Claims
Office Code for RMS output:

'ALL3233' and '2040' Santa Ana/Tustin, California
'ALL3221' and '1340' Torrence, California

.....and Format Names/IDs::

'APPD01'
'ADJD01'
'ADJD02'
'AUTD01'
'CAND01'
'CBKC01'
'CBKD01'
'CRED01'
'COMD01'
'ELCD01'
'ERRD01'
'IEBD01'
'IEBH01'

ARMS Process Report

'IEBT01'
'INSD01'
'OFFD01'
'PMTD01'
'RATD01'
'REND01'
'RMTD01'
'RNTD01'
'RNTD02'
'RPRD01'
'SURD01'
'VEDD01'

@Improvement Opportunities:

1.) Convert from standard ILE RPG program to an ILE program with multiple ILE RPG modules or service programs.

2.) Delete the 'AM0046V1 ' constant and use the Program Status Data Structure's left-justified PROGRAM ID value for the same purposes.

3.) There is hard-coded logic to check IF ARMS Trading Partner Insurance Company Insurance Company customer authorization extension ('EX') transaction set AND the adjuster or BCO change that is different than the centralized ARMS rental transaction database Adjuster Detail (AMADJD) file data, then to automatically generate a "pseudo-" Authorization ('AT') Change transaction set with only an adjuster detail (ADJD01) record. However, since this is information that is only for a specific X12 EDI standard compliant trading partner, that this special logic should be removed from this program and externally profiled in the EDI Translation and Interface/Mapper software to cause generation of the additional 'AT' Change when an Extension/Termination transaction set is received. Additionally, this EDI Translation and Interface Mapper software could execute an ARMS centralized database retrieval inquiry service utility program to retrieve specified information from the AMADJD file. Then this interface/mapper software could determine if this transaction set's adjuster information is different than the existing database file's information and if true, generate the additional 'AT' Change. This would add an 'AT' Change functional group transaction set to be output, along with the 'EX' from the same EDI transmission envelope, and it would add to the transaction load between AM0010V1 and AM0046V1 programs, inclusive. However, the benefit of standardizing the transaction set processing within AM0046V1 and moving profile-specific logic to the EDI translation interface/mapping software would be greater than having hard-coded within AM0046V1 program.

4.) There is currently an undesirable situation that occurs within this program. When any trading partner sends an authorization extension/termination transaction set, the adjuster detail is required to add callback detail note comments to the rental application database with the name of the adjuster changing the open rental contract/ticket's CURRENT AUTHORIZATION DATE field value. However, the adjuster making this change in this value may not be the claims adjuster assigned to manage the direct billing payment authorization for this rental contract. The adjuster making the extension/termination may be another adjuster taking on the extensions temporarily when the primary assigned adjuster is unavailable (i.e., day off due to vacation, illness, etc.), but that the primary assigned claims adjuster "of record" was not intended to be changed. Currently, the adjuster information in the Bill-To information within the rental contract is not updated by an extension/termination transaction set's adjuster information, but the centralized ARMS database adjuster detail (AMADJD) file is. This is erroneous to update AMADJD by an extension and this logic should be removed from AM0046V1 program. (A new issue will be submitted by Gina Miller, ARMS Technical Mgmt. staff to request this change.

ARMS Process Report

However, if AM0046V1 program's extension/termination logic is revised, then this issue should be considered.)

5.) Drop the constant detailed below and replace with attributes/field values within the ARMS Trading Partner Profile's Branch Claims Office / Enterprise Customer ID Cross-Reference (AMXBCO) file. Likewise, add the maintenance of these fields to the AMXBCO maintenance program and screen display file record format (AAAM10).

Process

Hierarchical numeric ID: 1.1.1.1.3.5
Coded name:
Name: DTQ Input to AM0025 (DQAM25V1)
Comment: @Definition: DQAM25V1 is a data queue used to provide input for PGM AM0025V1 which validates transaction set record formats.

Process

Hierarchical numeric ID: 1.1.1.1.3.6
Coded name:
Name: DTQ Input Data Queue to AM0030 (DQAMSET1)
Comment: @Definition: DQAMSET1 is a data queue used to provide input for PGM AM0030V1 which validates transaction set record formats data types.

Process

Hierarchical numeric ID: 1.1.1.1.3.7
Coded name:
Name: DTQ Input Data Queue to AM0040 (DQANDTL1)
Comment: @Definition: DQANDTL1 is a data queue used to provide input to PGM AM0040V1 which validates transaction set record format data.

Process

Hierarchical numeric ID: 1.1.1.1.3.8
Coded name:
Name: DTQ Input Data Queue to AM0046 (DQAM46V1)
Comment: @Definition: DQAM46V1 is a data queue used to provide input to PGM AM0046V1 which update the ARMS database and routes the transaction.

Process

Hierarchical numeric ID: 1.1.1.1.3.9
Coded name:
Name: DTQ Input to Program AM0100 (DQAMAP1)
Comment: @Definition: DQAMAP1 is a data queue used to input data to PGM AM0100 which distributes routed transactions from the centralized host processing machine.

Process

Hierarchical numeric ID: 1.1.1.1.3.10
Coded name: AMPSSR
Name: PGM Perform Internal Error Paging and Messaging (AMPSSR)

ARMS Process Report

Comment: @Definition: This is the ARMS *PSSR program exception error paging & messaging program (written in RPG/400 - Original Program Model).

@Purpose: This program's purpose is to output the error occurrence to the AMPGMERR Program Error Log file and if the error is not from a program executing on the "DEV" (development) machine, then to notify the ARMS On-Call and Backup persons via sending a pager message and sending an 'e-message' via call to MS0010 program of the platform name, program name and the error occurrence. This information will aid the ARMS On-Call person responding to resubmit any needed never-ending program job into the ARMS subsystem on the appropriate host computer system. This information will also cause them to research the dump taken in the previous program for the incompletely processed transaction specific information that it had been processing when the program halted or abnormally ended.

@Operational Method: This program outputs the passed entry parameters' values error occurrence to the AMPGMERR Program Error Log file. If the error is not on the "DEV" (development) host platform, then sending a pager message to the ARMS Primary and Backup On-Call pagers and sending an 'e-message' via call to MS0010 program of the platform name, program name and the error occurrence.

@Files: (CRUD)

- AMPERT (-R--) , accessed by AMPERTL1 logical access view.
- ANDQER (-R--) , accessed by ANDQERL2 logical access view.
- AMPGMERR (C---)
- FILE80 (C---)

@Notes: This program is used within every program's *PSSR program exception subroutine.

Process

#Hierarchical numeric ID: 1.1.1.1.3.11

#Coded name: CCDATE

#Name: PGM Validate Date and Convert Format (CCDATE)

Comment: @Definition: CCDATE is a date validation, conversion and manipulation callable OPM RPG program.

@Purpose: Executed from many ARMS programs for many purposes. The most common purposes are: to validate dates; convert formats; derive a future date by computing the adding provided days to a given date value; or, to derive the age of a renter by computing the number of years between the current date and a renter's date of birth. In a majority of programs, this program is executed to retrieve the current system date or timestamp in the MMDDYYYY or MMDDYY format and reformat it to YYYYMMDD format, or, to retrieve the day of week, such as required in the ARMS-to-ARMS Trading Partner Company Communication Sender / Receiver (AM***COM) program.

Validation is done throughout many ARMS programs, such as in the Validate Transmission Envelope (AM0010V1) program to validate the Transmission Date or in the Validate Transaction Set Record Format Data Types (AM0030V1) program's Edit Dates subroutine specifically to validate the date values for the following record formats' date value fields:

Authorization Detail's (AUTD01) Authorization Expiration Date, Date of Birth and Pickup Date

Renter Detail's (RNTD02) Date of Loss

Insured Detail's (INSD01) Policy Expiration Date

Cancellation Detail's (CAND01) Cancellation Date

Callback Detail's (CBKD01V1) Callback Origination Date

ARMS Process Report

Extension/Termination Detail's (CRED01) Termination Date and Renter
Notified Date
Rental Detail's (REND01) Rental Start and End Dates
Rental Segment Detail's (RATD01) Effective Dates of Rental Segments 1
through 4
Invoice Header's (IEBH01) Billing Start and End Dates
Payment Detail's (PMTD01) Payment Date

@Operational Method:

Before the CCDATE is called, the parameter fields must be set by the calling program. It is recommended that you do a clear the DTPARM external data structure subfields before any field values are loaded to insure that all fields have been initialized.

If a date conversion from MMDDYY to YYMMDD format, simply move the input date value to the From-Date (DTFRDT) input field after all the fields have been cleared and execute the CCDATE program. The outputted To-Date returned in DTTODT will be in YYMMDD format.

The following fields will be output by CCDATE, so it is not necessary that you fill them in. They are: DTFMTD, DTWKDY, DTMNTH, DTSTNG, DTMSID, and DTMSTX.

DTDAYS is not necessary for action code 'C' and will be zeroed out by CCDATE. DTDAYS will be filled in by the CCDATE program's execution when the action code is 'D'.

However, since AM0030V1 is requiring only date value validation, the program is executed after loading the following parameter subfield values in the following manner after all subfields were cleared/initialized:

Load 'C' (Convert) to ACTION CODE (DTACCD)
Load the DATE TO VALIDATE (in YYYYMMDD format) field value to FROM-DATE (DTFRDT)
Load the number 7 (YYYYMMDD) to the FROM-DATE-FORMAT (DTFRFT)
Load the number 6 (DDMMYYYY) to the TO-DATE-FORMAT (DTTOFT)
Execute the program
Check the contents of the Error Message ID (DTMSID). IF not blanks, then the FROM-DATE field value was invalid.

@Notes:

Documentation for usage can be found on the AS/400 in the ELLIB/QEISSRC/TLS.DAT text source member.

NEVER Use the DATE MULTIplied. by 1000.01 or 100.0001 to CONVERT! (Lack of precision and high overhead mathematical computation overhead.)

It should do hopefully everything anyone would need from such a program for any date value. Below is the explanation of its single parameter and how to use the program.

The parameters have been set up as in a empty database file called DTPARM that is used as an externally defined data structure on an input specification on the program's source code that causes the compiler to copy the subfield names, attributes, and starting positions. This is done so that a programmer can copy the parameters in as an external data structure instead of having to remember what order the parameters should be in and what length the fields are. The explanation of the external data structure DTPARM parameter data elements (subfields) passed as a single parameter field are:

ARMS Process Report

DTACCD - (1) - Processing Action Code possible values: 'C' -
 Convert (DEFAULT)

'D' - Compute the Days Difference between inputted dates

'A' - Add inputted Days to inputted From-Date
 DTFRDT - (8,0) - From-Date (DEFAULT is UDATE)
 DTFRFT - (1,0) - From-Date Format possible values: 1
 - MMDDYY (Month, Day, Year) (DEFAULT)

2 - DDMMYY (Day, Month, Year)

3 - YYMMDD (Year, Month, Day)

4 - YYDDD (Julian format)

5 - MMDDYYYY

6 - DDMMYYYY

7 - YYYYMMDD

8 - YYYYDDD (Julian format)

9 - MM00YY (Use this format to have the program determine the last day of the
 specified
 month and format it based on the To format.)

*NOTE: This format is only valid for this From-format!

DTTODT - (8,0) - To-Date
 DTTOTF - (1,0) - To-Date Format possible values: 1
 - MMDDYY (Month, Day, Year)

2 - DDMMYY (Day, Month, Year)

3 - YYMMDD (Year, Month, Day) (DEFAULT)

4 - YYDDD (Julian format)

5 - MMDDYYYY

6 - DDMMYYYY

7 - YYYYMMDD

8 - YYYYDDD (Julian format)
 DTDAYS - (7,0) - Number of Days to Add/Subtract or the computed
 Days Difference (+/-)
 DTFMTD - (10) - Displayed Formatted Date
 DTSEP - (1) - Display Separator (used in DTFMTD, i.e.,
 backslash, hyphen, or period)
 DTWKDY - (9) - Day of Week (Sunday, Monday, etc.)
 DTMNTH - (9) - Month Name (January, February, etc.)
 DTSTNG - (30) - Formatted Date String possible values:
 'C' - Month Day, Year (i.e., January 01,
 1992)

'F' (DEFAULT) - Day-of-week Month Day, Year (i.e., Wednesday January 01, 1992)

ARMS Process Report

DTMS - (7) - Error Message ID
DTMSTX - (80) - Error Message Text

@Improvement Opportunity:

The execution of this (CCDATE) program would be unnecessary if all of the ARMS programs that currently use this program were converted from the OPM RPG/400 to ILE RPG/IV program and these input date fields were attempted to be loaded into date type work fields. Also, any unsuccessful operation would need to be monitored for specific error exception types.

Process

Hierarchical numeric ID: 1.1.1.1.3.12
Coded name: AM2010V1
Name: PGM Retrieve Cross-reference (AM2010V1)
Comment: @Definition: This program is the callable ARMS generic Cross-Reference File Record Inquiry program.

@Purpose: To retrieve the associated ARMS Rental Cross-Reference File record when passed one of the following four input combinations: the ARMS VENDOR TRANSACTION ID; the ARMS TRADING PARTNER COMPANY PROFILE ID and the CUSTOMER TRANSACTION ID; the RENTAL LOCATION and the RENTAL CONTRACT/TICKET ID; or, the RENTAL LOCATION and the RENTAL BRANCH RESERVATION ID.

@Operational Method:

1.) The RENTAL LOCATION is always required and either a non-blank TICKET ID or RESERVATION ID must be passed.

IF RENTAL LOCATION, TICKET ID and RESERVATION ID are blanks, then load the RETURN CODE with 'E' (Error).

2.) IF Rental Location and Reservation ID are passed with the program call, use AMXREFL2 to retrieve the requested ARMS Cross-Reference File record.

a.) IF Ticket number is also passed with the program call, retrieve the record using AMXREFL2. Then validate the Ticket number passed with the program call against the Ticket data field XATKNO.

b.) IF passed TICKET ID and retrieved TICKET ID are different, then load the RETURN CODE with 'T' (Ticket ID Mismatch) and return to the calling program.

c.) IF the passed TICKET ID is same as the TICKET ID found for the reservation in the ARMS Cross-Reference File, then update all fields in the parameter list from AMXREFL2, set the RETURN CODE to blanks (blanks indicate successful) and return to the calling program.

3.) IF non-blank RENTAL LOCATION and TICKET ID are passed with the program, use AMXREFL1 to retrieve the data.

IF no record found for RENTAL LOCATION and TICKET ID, then load the RETURN CODE with 'E' (Error) .

4.) If an ARMS COMPANY PROFILE ID is passed with the program call and a record is retrieved, compare this value with XACUID in AMXREFLn.

IF they are different, change the RETURN CODE to 'P' for 'Profile Error' and return to calling program.

5.) IF non-blank ARMS COMPANY PROFILE ID and CUSTOMER TRANSACTION ID are passed, retrieve record from AMXREFL3.

a.) IF TICKET ID / RESERVATION ID is passed, compare these with values retrieved from AMXREFL3.

IF they do not agree, change return code to 'T' / 'R'.

b.) IF RENTAL LOCATION is passed, compare this value with value retrieved from AMXREFL3.

IF they do not match, change return code to 'L' .

6.) IF an ARMS VENDOR TRANSACTION ID is passed, retrieve record from AMXREF.

a.) IF an ARMS COMPANY PROFILE ID is passed with the program call and a record is retrieved, compare this value with the associated field in the retrieved ARMS Cross-Reference File.

IF they are different, change the return code to 'P' for 'Profile Error' and return to calling program.

b.) IF RENTAL LOCATION is passed, compare this value with the AMXREF value. IF they are different, change return code to 'L'.

c.) IF TICKET ID or RESERVATION ID is passed, compare these values with values retrieved from AMXREF.

IF they do not agree, change RETURN CODE to 'T' / 'R'.

7.) IF a record is retrieved from AMXREF or any of these logical files based upon the file use hierarchy, and additional data fields are passed, and more than one data mismatch occurs, change the return code to 'M' for multiple errors.

Hierarchy for record retrieval:

1.) IF ARMS VENDOR TRANSACTION ID is passed, and any other data fields, retrieve from AMXREF.

2.) IF ARMS COMPANY PROFILE ID and CUSTOMER TRANSACTION ID are passed, and any other data fields (except VENDOR TRANSACTION ID), retrieve from AMXREFL3.

3.) IF RENTAL LOCATION and RESERVATION ID are passed, retrieve from AMXREFL2.

4.) IF RENTAL LOCATION and TICKET ID are passed, retrieve from AMXREFL1.

RETURN CODE possible values:

blank : Successful retrieval / edit
L : Location passed is blank / invalid
: Ticket/Reservation passed is blank
T : Ticket passed, unmatched on AMXREF record
P : Profile passed, unmatched on AMXREF record
E : No AMXREF record found
M : Multiple errors

ARMS Process Report

@Notes: No exception error handling is specified.

This executable program is executed by many other ARMS programs that are processing a ARMS transaction request to retrieve information from the ARMS Rental Cross-Reference File. This program is currently called with parameters by the following programs: AMLCXLTL, AMRTVSQ, AM0020V1, AM0021V1, AM0030V1, AM0040V1, AM0070V1, AM0120, AM0129, AM0150, AM0900, AM1080V1, AM1085V1, AM5010, AM5015, and CCAM16. The parameters used in calling this program are the following:

	Output	1 character	RETURN CODE
Input/Output	20 character		VENDOR TRANSACTION ID
Input/Output	5 character		ARMS TRADING PARTNER COMPANY PROFILE
ID			
Input/Output	20 character		CUSTOMER TRANSACTION ID
Input/Output	10 character		RENTAL LOCATION
Input/Output	6 character		RESERVATION ID
Input/Output	6 character		TICKET ID
	Output	1 character	STATUS CODE
	Output	8/0 numeric	STATUS CODE LAST CHANGED DATE
	Output	6/0 numeric	STATUS CODE LAST CHANGED TIME
	Output	10 character	STATUS CODE LAST CHANGED PROGRAM
ID			
	Output	1 character	MACHINE ID
	Output	10 character	BRANCH CLAIMS OFFICE CODE
	Output	1 character	RENTAL SYSTEM SOURCE ID
	Output	8/0 numeric	RECORD ADD DATE
	Output	6/0 numeric	RECORD ADD TIME
	Output	10 character	RECORD ADD PROGRAM ID
	Output	8/0 numeric	RECORD CHANGE DATE
	Output	6/0 numeric	RECORD CHANGE TIME
	Output	10 character	RECORD CHANGE PROGRAM ID

@Files: (CRUD)

- AMXREF (-R--) ARMS Cross-Reference File, keyed by Vendor Transaction ID

- AMXREFL3 (-R--) ARMS Cross-Reference File, keyed by Company Profile ID and Customer Transaction ID

- AMXREFL1 (-R--) ARMS Cross-Reference File, keyed by Rental Location ID and Rental Ticket ID

- AMXREFL2 (-R--) ARMS Cross-Reference File, keyed by Rental Location ID and Rental Reservation ID

@Improvement Opportunities:

Replace the internally defined parameter fields in this program and all programs executing this program with a database file with no record to be used as an externally defined data structure so that only one definition of the parameter's data elements exist, along with comments on their purpose and possible values.

Replacing the execution of this (AM2010V1) program with the ARMS Inquire/Update Cross-Reference File record (AM1010V1) program in many other programs if it is called in "Inquiry" mode.

Process

Hierarchical numeric ID: 1.1.1.1.3.13

ARMS Process Report

Coded name: AM0090V1

Name: PGM Generate External Error (AM0090V1)

Comment: @Definition: This program is the ARMS called External Error Handler Program.

@Purpose: To generate a single Error (ER) group type transaction data set to be sent to the Rental Management Trading Partner to indicate that the ARMS host rejected a transaction data set received from their host application system. The error is also logged to an error log file.

@Operational Method:

- Verifies the passed input parameter error reason codes (maximum 20) that are not equal to '99' against the ARMS Error Code and Description Table (AMERRTBL) file for valid error codes and returns an "A" for accepted or an "R" for rejected. If the passed error reason code does not exist in this file, the ARMS Program Exception Handler (AMPSSR) program is executed to notify the ARMS On-Call staff of the program-to-program referential discrepancy and the erroneous error code value is concatenated with the text "ERROR CODE IS INVALID" and this resulting value is written to the ARMS Error Log (AMERRLOG) file record's FILLER field and its ERROR CODE field is loaded with the value "99".

- Write Error (ER) group type transaction data set records to one of two possible transaction data set output files. If the program is being executed on the centralized ARMS host computer system platform, it will write the output record formats to the ARMS Transactions To Be Packaged (AMPACK) transaction file as centralized transactions to be packaged for sending. If this program is executing on the distributed rental computer system platform, then it will write the output record formats to the ARMS Distributed Transactions To Centralized Input (ANDIST) transaction file for the distributed system transaction to be sent to the centralized system.

- Sends the written Error group type transaction set record formats' key value to the output file's associated data queue. If the records were output to the AMPACK file, then the key is sent its associated Transaction Packaging Input (DQAMPKG) data queue. If the records were output to the ARMS (ANDIST) file, then the key is sent its associated Transaction Distributed-to-Centralized Input (DQANDST) data queue.

- Before exiting the program, a record is also added to rejected transactions error log file (AMERRLOG). that will appear on the ARMS Error Log Report (CLL817/AM5000) and the ARMS Error Log Inquiry Screen (CCAM14).

@Notes: This program is only executed by other ARMS never-ending programs that have validated a received transaction data set as being in error and rejected it from further processing to eventually update the Rental Systems application database. This program is currently called with parameters by the following programs: AM0020V1, AM0021V1, AM0025V1, AM0030V1, and EC00EXV1. (NOTE: This program has been replaced in many other programs that it had been called from by calling the newer ARMS Internal/External Error Handler 'AM0098' program.) The parameters used in calling this program are the following:

Input	5 character	COMPANY ID
Input	9/0 numeric	TRANSMISSION CONTROL ID#
Input	5/0 numeric	GROUP CONTROL ID#
Input	2 character	GROUP TYPE CODE
Input	20 character	CUSTOMER TRANSACTION ID
Input	20 character	VENDOR TRANSACTION ID
Input	6 character	FORMAT ID IN ERROR

ARMS Process Report

Input 40 character ERROR REASON CODE (2 characters per code/20 occurrences maximum)
 Output 1 character PROGRAM RETURN CODE (Possible values: R=Rejected - default, or A=Accepted)
 Input 10 character PROGRAM NAME IN ERROR (Rejecting the transaction processing)

@Files: (CRUD)

- AMPACK (C---)
- ANDIST (C---)
- AMERRTBL (-R--)
- AMERRLOG (C---)

@Embedded Data: (Constants)

"ERROR CODE IS INVALID" as suffix for invalid error code.

@Improvement Opportunity:

Since the text is the same in all of the files, move the constant's error message text into an added record in the AMERRTBL file with the ERROR CODE value of blanks that this program, AM0090V1 Generate ARMS External Error and AM0098 ARMS Error Handler program could all retrieve.

Process

Hierarchical numeric ID: 1.1.1.1.3.14

Coded name: AM0097V1

Name: PGM Handle Internal Error (AM0097V1)

Comment: @Definition: This program is the ARMS called Internal Error Handler Program.

@Purpose: To notify the ARMS Application Development Department On-Call personnel of a transaction error condition that requires research and possible manual intervention. Error (ER) group type transaction data set in response to a rejected transaction data set received from a Rental Management Trading Partner host application system and to log the error to an error log file.

@Operational Method:

- Verifies each non-blank passed input parameter error reason code that is not equal to '99' or '00' exists in the ARMS Valid Error Codes table file (AMERRTBL) and returns an 'A' for accepted or an 'R' for rejected. If the passed error reason code does not exist in this file, the ARMS Program Exception Handler (AMPSSR) program is executed to notify the ARMS On-Call staff of the program-to-program referential discrepancy and the erroneous error code value is concatenated with the constant text "ERROR CODE IS INVALID" and this resulting value is written to the ARMS Reservation Transfer (and Outbound) Errors Log (AMXFRLOG) file record's FILLER field and its ERROR CODE field is loaded with the value '99' (Unknown).

- If the passed CALLING PROGRAM ID is NOT 'AM0046V1', then execute call to the ARMS Program Exception Handler 'AMPSSR' program, passing the CALLING PROGRAM ID and the Transaction's 96-character key value that is comprised of the passed TRANSMISSION CONTROL ID#, GROUP CONTROL ID#, GROUP TYPE CODE, CUSTOMER TRANSACTION ID and VENDOR TRANSACTION ID, padded with blanks.

@Note: This program is only executed by other ARMS never-ending programs or called programs that cannot process a ARMS transaction request as being in error

ARMS Process Report

or not doable and reject it from further processing. This program is currently called with parameters by the following programs: AM0025V1, AM0030V1, AM0047V1, AM0062V1, AM1080V1, AM2090V1, CC2090V1 and EC2090V1. (NOTE: This program has been replaced in many other programs that it had been called from by calling the newer ARMS Internal/External Error Handler 'AM0098' program.) The parameters used in calling this program are the following:

Input	5 character	COMPANY ID
Input	9/0 numeric	TRANSMISSION CONTROL ID#
Input	5/0 numeric	GROUP CONTROL ID#
Input	2 character	GROUP TYPE CODE
Input	20 character	CUSTOMER TRANSACTION ID
Input	20 character	VENDOR TRANSACTION ID
Input	6 character	FORMAT ID IN ERROR
Input	40 character	ERROR REASON CODE(S) (2 characters per
code/20 occurrences maximum)		
Output	1 character	PROGRAM RETURN CODE (Possible values:
R=Rejected - default, or A=Accepted)		
Input	10 character	PROGRAM NAME IN ERROR (Rejecting the
transaction processing)		

@Files: (CRUD)

- AMERRTBL (-R--)
- AMXFRLOG (C---)

@Embedded Data: (Constants)

"ERROR CODE IS INVALID" as suffix for invalid error code.

@Improvement Opportunity:

Since the text is the same in all of the files, move the constant's error message text into an added record in the AMERRTBL file with the ERROR CODE value of blanks that this program, AM0090V1 Generate ARMS External Error and AM0098 ARMS Error Handler program could all retrieve.

Process

Hierarchical numeric ID: 1.1.1.1.3.15

Coded name: CRTF80

Name: PGM Create MS01 (CRTF80)

Comment: @Definition: This is the executable OPM CL program to Create and Clear the temporary Send "e-Messages" Work (FILE80) Message Detail Lines file.

@Purpose: To duplicate, if necessary, and clear a temporary output work file prior to outputting "e-message" detail message lines from an executing program to be used as input to a subsequent execution of the Send "e-Message" (MS0010) program.

@Operational Method:

Checks the existence of a ISS: SM0003 80-position work file (FILE80) in this active job session's QTEMP Temporary Job Session library.

IF the file does not exist, then retrieve the library from this file from any library within this job session's library list. Then create a duplicate copy of this file in the QTEMP Temporary Job Session library without copying any data within the file.

ARMS Process Report

Clear the 80-position work file (FILE80) in this active job session's QTEMP Temporary Job Session library.

Execute an override of any specified 80-position work file (FILE80) in this active job session for the scope of the entire job to 80-position work file (FILE80) in this active job session's QTEMP Temporary Job Session library and sequentially block its access with 409 records per block.

@Notes:

This program was created and is stewarded by the Information Systems Support Department's ISA Group.

No parameters are used in the execution of this program.

@Files: (CRUD)

FILE80 (----) File is created in the QTEMP library if necessary and then cleared.

Process

Hierarchical numeric ID: 1.1.1.1.3.16

Coded name: RTVMSGTS

Name: PGM Retrieve Message Status (RTVMSGSTS)

Comment: @Definition: This is an executed OPM CL program to retrieve the status of the Enterprise Rent-A-Car Message Application System.

@Purpose: To retrieve the status of the Enterprise Rent-A-Car Message Application System before execution of the Send a Message (MS0010) program.

@Operational Method:

Load the default value 'ER' (Error) to STATUS CODE.

IF currently executed on the "DEV" machine, then retrieve production objects from ELPRD library instead of the ELLIB library.

Retrieve the logical value of the MESSAGE SYSTEM STATUS from its Status data area (DAMSGSTS).

IF the MESSAGE SYSTEM STATUS is "ON" ('1' or True), then the MESSAGE SYSTEM is UP. Then check the DDM and OPTICONNECT/400 STATUS values on the 'OPTICON' machine/system platform. IF both are OK, then pass 'OK' to STATUS CODE.

IF the MESSAGE SYSTEM STATUS is "OFF" ('0' or False), then the MESSAGE SYSTEM is DOWN.

Check if the Temporary Messages Hold Files (MSHDR, MSTXT, MSNOTIFY) are exist and in which library they were found. IF they do exist and were found in the ELSYSOVR library, then pass 'TF' to STATUS CODE.

@Notes:

This program was created and stewarded by the Information System Support (ISS) Department's ISA-Applications Group.

ARMS Process Report

This program is currently executed by the following programs prior to the execution of the Send e-Message program:

ARMS *PSSR Program Error Paging & Messaging (AMPSSR) program
ARMS Application Edit/Router (AM0040V1) batch program
ARMS Non-Inbound Format Data Replacement Alteration (AM0047V1) program
ARMS Internal/External Error Handling (AM0098) program

The program is executed with one output parameter value of a 2 character STATUS CODE that has one of the following values:

'OK' - Message System is UP (available) and DDM Connections Tested.
'TF' - Message System is DOWN (unavailable), but the Temporary Messages Hold Files are in place to receive messages that are being requested to be sent.
'ER' - Either Message System is down and the Temporary Messages Hold Files are NOT in place to receive messages that are being requested to be sent or DDM Connection FAILED.

@Files: (CRUD) (All are checked for existence and which in library they were found.)

MSHDR - Message System Message to be Sent - Header/Abstract
MSTXT - Message System Message to be Sent - Detail Text Lines
MSNOTIFY - Message System Message to be Sent - Users/Distribution

Process

Hierarchical numeric ID: 1.1.1.1.3.17

Coded name: SNDEMSG

Name: PGM Send Enterprise Message (SNDEMSG)

Comment: @Definition: This is the executable ILE RPG command for sending enterprise message

@Purpose: To send previously outputted "e-message" to the specified ARMS On-Call Message Distribution List or the ARMS Technical Management from an another program.

@Operational Method:

Using the QTEMP library temporary work files as input for the formatted message, read the Message Program - Send Msg to employees file to obtain the message distribution.

@Notes:

This program was created and stewarded by the Information System Support (ISS) Department's ISA-Applications Group.

Process

Hierarchical numeric ID: 1.1.1.1.3.18

Coded name: AM0098

Name: PGM Handle Internal/External Error (AM0098)

Comment: @Definition: This is the newer ARMS Internal/External Error Handling Program. This program performs functions beyond both the AM0090V1 and AM0097V1 program.

ARMS Process Report

@Purpose: To notify the ARMS Application Development Department On-Call personnel of a transaction error condition that requires research and possible manual intervention. Error (ER) group type transaction data set in response to a rejected transaction data set received from a Rental Management Trading Partner host application system and to log the error to an error log file.

(NOTE: This program will replace AM0090V1 and AM0097V1 with the new error handling process. Error records are written to the error files AMERRLOG and ANDQER and transaction files ANDIST or AMPACK depending on what machine the program is currently running on. The program currently contains both the old and new error handling logic.)

@Operational Method:

- Verifies each non-blank passed input parameter error reason code that is not equal to '99' or '00' exists in the AMERRTBL Valid Error Codes table file and returns an 'A' for accepted or an 'R' for rejected. If the passed error reason code does not exist in the AMERRTBL file, AMPSSR program is executed to notify the ARMS On-Call staff of the program-to-program referential discrepancy and the erroneous error code value is concatenated with the constant text "ERROR CODE IS INVALID" and this resulting value is written to the AMXFRLOG file record's FILLER field and its ERROR CODE field is loaded with the value '99' (Unknown).

- If the passed CALLING PROGRAM ID is NOT 'AM0046V1', then execute call to 'AMPSSR' passing the CALLING PROGRAM ID and the Transaction's 96-character key value that is comprised of the passed TRANSMISSION CONTROL ID#, GROUP CONTROL ID#, GROUP TYPE CODE, CUSTOMER TRANSACTION ID and VENDOR TRANSACTION ID, padded with blanks.

@Notes: This program is only executed by other ARMS never-ending programs or called programs that cannot process a ARMS transaction processing request as being in error or not doable and rejected it from further processing. This program is currently called with parameters by the following programs: AM0040V1, AM0046V1, AM0060V1, AM0061V1. (NOTE: This program is yet to replace the call to AM0097V1 older Internal Error Handler program in many other programs: AM0025V1, AM0030V1, AM0047V1, AM0062V1, AM1080V1, AM2090V1, CC2090V1 and EC2090V1.) The parameters used in calling this program are the following:

Input	5 character	COMPANY ID
Input	9/0 numeric	TRANSMISSION CONTROL ID#
Input	5/0 numeric	GROUP CONTROL ID#
Input	2 character	GROUP TYPE CODE
Input	20 character	CUSTOMER TRANSACTION ID
Input	20 character	VENDOR TRANSACTION ID
Input	6 character	FORMAT ID IN ERROR
Input	40 character	ERROR REASON CODE(S) (2 characters per
code/20 occurrences maximum)		
Output	1 character	PROGRAM RETURN CODE (Possible values:
'R'=Rejected - default, or 'A'=Accepted)		
Input	10 character	PROGRAM NAME IN ERROR (Rejecting the
transaction processing)		
Input	150 character	Transaction key value
Input	128 character	Transaction data value
Input	10 character	Program ID that is rejecting transaction
Input	4 character	Enhanced Error Type Code
Input	10 character	Data Queue Name that provided rejected
transaction key as input.		
Input	1 character	Error Severity flag

ARMS Process Report

Input 1 character Error Transaction type flag (possible values: 'O'=Outbound, 'I'=Inbound, and 'T'=Transfer) (currently not referenced by this program)

Input	8 character	Alpha Add Date (ccyyymmdd format)
Input	6 character	Alpha Add Time (24-hour format)
Input	10 character	Calling Program ID
Input	5 character	Add Employee ID

@Files: (CRUD)

- AMPACK (C---)
- ANDIST (C---)
- ANDQER (CRU-)
- AMPERT (-R--)
- AMERRTBL (-R--)
- AMERRLOG (C---)

@Constants:

"ERROR CODE IS INVALID" as suffix for invalid error code.

'RTVMSGSTS' is the called program to check if the Message System Status

is active.

'ARMS TEST MESSAGE' is the Test Message Header Text that is not referenced within the program.

Process

Hierarchical numeric ID: 1.1.1.1.3.19

Coded name: AM2050V1

Name: PGM Retrieve Machine ID (AM2050V1)

Comment: @Purpose: To return the associated MACHINE ID (host platform name) for a given rental location.

@Operational Method:

- This program attempts to retrieve a branch information record from the Office Directories Branch Office (OFFDRB) file for the passed GROUP ID and BRANCH ID specified as input in the parameter list.

- IF both the passed GROUP ID and BRANCH ID are blank, then return '***ERROR' as MACHINE ID to the calling program.

- IF the passed BRANCH ID is blank, then attempt to retrieve the first record associated for the passed GROUP ID.

..IF found, then return its MACHINE ID value.

IF the passed BRANCH ID is not blank, then attempt to retrieve the record associated for the passed GROUP ID and BRANCH ID values.

..IF branch is found, then return its MACHINE ID value for that branch office location record to the calling program.

@Notes:

Executed with the following parameters:

Input	2 character	GROUP ID
Input	2 character	BRANCH ID
Output	8 character	MACHINE ID

ARMS Process Report

@Embedded Data/Constants:

'***ERROR' is the constant value loaded if no OFFDRB file record is retrieved successfully.

Process

Hierarchical numeric ID: 1.1.1.1.3.20

Coded name: AM2051V1

Name: PGM Check for Holiday / After Hours (AM2051V1)

Comment: @Definition: This program determines if a given rental location is not able to accept new ARMS authorized reservations and detours it to Claims Connection .

@Purpose: To return the associated MACHINE ID (host platform name) for a given rental location's passed GROUP ID and BRANCH ID.

@Operational Method:

- Receives three (3) input parameters:
2 character GROUP ID
2 character BRANCH ID
1 character ARMS Profile ID "Send to Claims After Hours?" Flag
- Returns a single output parameter:
8 character MACHINE ID (NAME)
- The following decision table/grid shows which location a new or transferred authorized reservation will be routed based on this flag setting and other criteria:

Active

Rental

Location		Send to Claims After hours	Holiday	Weekend	Claims	Branch	Routed Open	Open
-----		-----						
N/A	Claims	Y		Y	N/A		N/A	N/A
	ECARS	Y		N	Y		N/A	
Y	Claims	Y		N	N		Y	
	ECARS	Y		N	N		Y	
N	Claims	Y		N	N		N	
	ECARS	Y		N	N		N	
Y	Claims	Y		N	N		N	
	ECARS	Y		N	N		N	
N	Claims	N		N/A	N/A		N/A	N/A
	ECARS	N		N/A	N/A		N/A	N/A

Retrieve the current system date, day of week and time (adjusted to that location's time zone).

ARMS Process Report

IF the passed input parameter value for the ARMS Profile flag "Send to Claims After Hours?" is "N" (No), then do the processing to determine if the passed GROUP ID and BRANCH ID is an active daily rental branch location.

Attempt to re-retrieve the associated record's BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION field in the Office Directory Branch Master (OFFDRB) file by the passed GROUP ID if the passed BRANCH ID is blank, or use GROUP ID and the BRANCH ID.

Also, IF a record exists for just the passed GROUP ID, then indicate that the branch supports daily rental business, ELSE use the retrieved OFFDRB file record's BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION field value.

Additionally, IF an OFFDRB file record was retrieved successfully AND its STATUS CODE is "A" (Active) and its BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION is "Y" (Yes), then load the MACHINE ID output parameter field with blanks. IF today is NOT a national holiday, then,

IF the associated OFFDRB file record was not successfully retrieved, OR the retrieve record's STATUS CODE is NOT "A" (Active), OR its BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION is NOT "Y" (Yes), then load the returned MACHINE ID output parameter field with '***ERROR'.

ELSE, load the MACHINE ID field with blanks

IF the passed location is not found, or not a daily rental business location, or its status is not active, then return "***ERROR" in the MACHINE ID parameter.

IF the passed input parameter value for the ARMS Profile flag "Send to Claims After Hours?" is "Y" (Yes), then do the processing to determine if today is a national holiday:

- Attempt to retrieve the associated Office Directory Branch Master (OFFDRB) file record using the passed GROUP ID and BRANCH ID.
- IF unsuccessful, then indicate that today is a holiday.
- IF successful, then using the retrieved OFFDRB file's MACHINE ID value access the associated ISS: Global Attributes (GACTL) File record to retrieve this branch's platform's associated COUNTRY NAME. Then use today's system date and this retrieved COUNTRY NAME value to attempt to access an existing record in the ECR: National Holiday (RA005P00) file. IF a record exists, then indicate that today is a national holiday.

Attempt to re-retrieve the associated record's BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION field in the Office Directory Branch Master (OFFDRB) file by the passed GROUP ID if the passed BRANCH ID is blank, or use GROUP ID and the BRANCH ID.

Also, IF a record exists for just the passed GROUP ID, then indicate that the branch supports daily rental business, ELSE use the retrieved OFFDRB file record's BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION field value.

Additionally, IF an OFFDRB file record was retrieved successfully AND its STATUS CODE is "A" (Active) and its BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION is "Y" (Yes), then load the MACHINE ID output parameter field with blanks. IF today is NOT a national holiday, then,

IF the associated OFFDRB file record was not successfully retrieved, OR the retrieve record's STATUS CODE is NOT "A" (Active), OR its BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION is NOT "Y" (Yes), then load the returned MACHINE ID output parameter field with '***ERROR'.

ELSE, load the MACHINE ID field with blanks.

ARMS Process Report

IF today is NOT a national holiday, then check the date to determine if its day of week is Saturday or Sunday.

IF not found, then Also, check if the today is a national holiday by checking for the existence of a record in the ECR: National Holiday (RA005P00) file by today's system date and the Country Name from retrieving the ISS: Global Attributes (GACTL) File by the MACHINE NAME from the retrieved Office Directory Branch Master (OFFDRB) file. This is to determine if the office is open and will accept new business.

IF it is a holiday, it is Saturday or Sunday, or the branch office is closed at the branch's current local time, then attempt to retrieve the Office Directory Branch File record for GROUP ID = "76" and BRANCH ID = "80" and check if this Claims Connection location is currently open with the current time adjusted to its local time zone. IF it is open, then return this GROUP ID and BRANCH ID, along with its current MACHINE ID value. IF it is not open, then return the original passed rental location's MACHINE ID for the previously retrieved closed branch office location record.

The ARMSPR1 Send to Claims After Hours flag (P1YN18) to determine whether to send a new reservation to Claims Connection or the closest Group/Branch. If this flag is set to 'N' for a trading partner, the new reservation will not be sent to Claims Connection whether it is a holiday or not. This will prevent authorizations for Canadian locations being sent to Claims Connections on U.S. national holidays.

@Files: (CRUD)

OFFDRB (-R--) Office Directories Branch Office File, keyed by GROUP ID and BRANCH ID.

GACTL (-R--) ISS: Global Attributes File

RA005P (-R--) ECR: National Holiday File

(Example: '19980701')

'CANADA' is the Canadian national holiday for their Independence Day.)

@Notes:

Executed with the following 4 input parameters:

Input	2 character	GROUP ID
Input	2 character	BRANCH ID
Input/Output	8 character	MACHINE ID (distributed application database host platform logical identifier/identifying name)
Input	1 character	SEND TO CLAIMS CONNECTION AFTER HOURS INDICATION FLAG

Currently only executed from the ARMS Application Edit/Router Batch (AM0040V1) program.

@Embedded Data/Constants:

'***ERROR' is the constant value loaded if no active daily rental business type Office Directory Branch Location (OFFDRB) file record is retrieved successfully.

'Mon', 'Tue', 'Wed', 'Thu', 'Fri' are used to associate the correct OFFDRB file record's Open Time and Close Time for each Day of the Week.

'Sat' and 'Sun' are constant values used to determine if the current day of the week is on a weekend.

ARMS Process Report

76 is the constant overriding value for the Claims Connection GROUP ID.
80 is the constant overriding value for the Claims Connection BRANCH ID.

@Improvement Opportunities:

Only attempt to retrieve the associated OFFDRB file record in a single routine. Currently done in both the \$OFFDR subroutine to Retrieve OFFDRB (Office Directory File) Record and the \$HOLDY subroutine to Check RACNTR to determine if Holiday.

Process

Hierarchical numeric ID: 1.1.1.1.3.21
Coded name: AM2053V1
Name: PGM Retrieve State Code (AM2053V1)
Comment: @Definition: This is the callable OPM RPG program to Retrieve State Code for a given Rental Location ID.

@Purpose: To extract and return the associated State Code from the Daily Rental Office Location Directory file for a given Rental Location ID.

@Operational Method:

Program accepts a two parameter list, composed of an input field, LOCATION ID (10 character), and an output field, STATE CODE (2 character). A record is attempted to be retrieved from Daily Rental Office Directory (DROFLF1) file with Location ID as its access key.

IF a record is found, the State Code will be extracted from the retrieved record's ADDRESS LINE 2 (CITY, STATE, ZIP) field value and will be loaded to the output parameter STATE CODE to be returned to the calling program. ELSE, set the STATE CODE to '**' and return to the calling program.

@Notes: This callable program is executed by the following ARMS programs:

ARMS Application Edit/Router Batch (AM0040V1)
ARMS Add Surcharge (AM0096V1)
ARMS Generate Underage Surcharge (AM1085V1)

@Files: (CRUD)

DROFF (-R--)

@Embedded Data/Constants:

'**' is the returned output STATE CODE if record is not retrieved.

@Improvement Opportunities:

1. Convert from OPM RPG/400 to ILE RPG/400 service program that is over all Office Directory files (internal and external organizations' rental locations) that will provide also Office Name, Street Address #1, Telephone Number and other information so that the ARMS application is separated from the Office Directories database access software tier.

Process

Hierarchical numeric ID: 1.1.1.1.3.22
Coded name: AM0047V1
Name: PGM Enforce rules of Data Exchange for Outbound Transaction (AM0047V1)

ARMS Process Report

Comment: @Purpose: To generically enforce the rules specified by the ARMS Rental Management Trading Partners for data exchange on transaction data sets sent to adjusters from Enterprise Rent-A-Car or Claims Connection. This can comprise replacing data in a transaction data set's record formats' data elements that were from the rental application database with values retrieved from the ARMS centralized database files' data that had been populated by the transactions received from the adjuster. This will then "echo" back information to the trading partner's application that had been sent to ARMS.

@Operational Method:

Using the passed TRADING PARTNER PROFILE ID and ARMS DATABASE FILE FIELD NAME, attempt to retrieve the associated ARMS Trading Partner File/Field Control Data (ARMSPR4) file.

IF record not found, return to calling program without loading the PROGRAM RETURN CODE parameter.

IF record is found, then based on the two flags in the trading partner profile, use the following decision table:

CASE 1 : FORMAT field is blank DATABASE field is NOT blank.

	Mandatory To Send	Allow Field Changes	Action	
	Y		N/A	assign
database field				to
format field error				
code 04, missing				
corrected from				
database				
	N		N	assign
database field				to
format field if				
rental status not				
Waiting and not				
Unconfirmed				

CASE 2 : BOTH fields are Non-Blank

	Mandatory To Send	Allow Field Changes	Action	
	N/A		N	assign
database field				to
format field if				
rental status not				
Waiting and not				
Unconfirmed				

CASE 3 : BOTH fields are Blank.

ARMS Process Report

Mandatory Send

Allow Field Changes

tion

Y

N/A

error

code 01, missing

could not correct

In all other cases, the format field remains unchanged regardless of the database field value.

@Notes:

This program is currently only called by the ARMS Database update and routing (AM0046V1) program with the following multiple parameters:

Input	50 character	ARMS Database File field value
Input/Output	50 character	Transaction Set Record Format field
Old/Replaced value		
Input	6 character	ARMS Database File Field Name
Input	1 character	ARMS Rental Cross-Reference File
Record's STATUS CODE		
Input	5 character	COMPANY PROFILE ID
Input	9/0 numeric	TRANSMISSION CONTROL ID
Input	5/0 numeric	GROUP CONTROL ID
Input	2 character	GROUP TYPE CODE
Input	20 character	CUSTOMER TRANSACTION ID
Input	20 character	VENDOR TRANSACTION ID
Input	6 character	FORMAT ID IN ERROR
Output	40 character	ERROR REASON CODES (20 occurrences maximum @ 2 characters, however, only first occurrence used.)
Output	1 character	PROGRAM RETURN CODE (Not loaded by this program, but passed on from executed AM0097V1 program.)
Input	10 character	CALLING PROGRAM ID IN ERROR

@Files: (CRUD)

ARMSPR4 (-R--) ARMS Trading Partner File/Field Control Data file
 FILE80 (C---) Send a Message 80-position work file (created empty and overridden to in the QTEMP library by the execution of 'CRTF80' program, then opened explicitly.)

@Embedded Data/Constants:

'RTVMSGSTS' is the constant defined in order to execute the program before calling the Send e-Message (MS0010) program.

The following compile-time array "SBJ" is composed of 10 constant text lines (47 characters each) used for loading e-message header and detail lines for any mandatory data to be sent that is missing from the record format and the ARMS database:

- 1.) MANDATORY TO SEND FIELD IS MISSING
- 2.) Format:
- 3.) Program:
- 4.) Company:
- 5.) Transmission Control ID:
- 6.) Group Type:
- 7.) Cust Tran ID:
- 8.) Vndr Tran ID:
- 9.) is missing-corrected from database

ARMS Process Report

10.) is missing - could not be corrected from database

@Improvement Opportunities:

1.) Convert this OPM RPG/400 program to an ILE RPG module and have this data replacement enforcement logic used as the transaction data set's record format's field are being loaded.

2.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.

Process

Hierarchical numeric ID: 1.1.1.1.3.23

Coded name: AM0096V1

Name: PGM Generate Pseudo-Authorization Change Transaction Data Set (Approved Govt.Surcharges) (AM0096V1)

Comment: @Purpose: To generate proprietary approved surcharge formats from the formats stored in the pre-approved surcharges file, AMSURTBL, based on the state and rental location.

@Operational Method:

- Delete all automatically generated (ORIGIN CODE = 'A') in Authorized Surcharge Detail file, AMSURD that are associated with the passed VENDOR TRANSACTION ID and generate AMMNTLOG ARMS Database Maintenance Log file records via call to AM1050V1.
- Pass the CURRENT RENTAL LOCATION ID to the called program AM2053V1 to retrieve the STATE CODE.
- Search AMSURTBL file for any records with BLANK COMPANY PROFILE ID whose LOCATION ID field value matches with the passed CURRENT RENTAL LOCATION Group ID or Group/Branch ID value. For every blank Location ID record or matches the passed CURRENT RENTAL LOCATION ID read, write an AMSURD Approved Surcharge Detail File record for that retrieved AMSURTBL file record's SURCHARGE CODE, SURCHARGE AMOUNT and SURCHARGE CHARGE FREQUENCY field values.
- Search AMSURTBL file for any records with the passed COMPANY PROFILE ID (that was embedded within the passed Transaction Data Set Key) whose LOCATION ID field value matches with the passed CURRENT RENTAL LOCATION ID Group ID or Group/Branch ID value. For every blank Location ID record or matches the passed CURRENT RENTAL LOCATION ID read, write an AMSURD Approved Surcharge Detail File record for that retrieved AMSURTBL file record's SURCHARGE CODE, SURCHARGE AMOUNT and SURCHARGE CHARGE FREQUENCY field values.
- Using the passed CURRENT RENTAL LOCATION ID Group/Branch ID, retrieve the associated BC026P00 file record. For any non-zero miscellaneous government surcharges that are applicable for this rental location, write an AMSURD Approved Surcharge Detail File record for that SURCHARGE AMOUNT and SURCHARGE CHARGE FREQUENCY field values loading the SURCHARGE CODE with '05' (Government).
- Return to calling program by explicit SETON *INLR.

@Notes: This program is called from AM0046V1 program when processing a transaction data set for the following conditions: an Authorization (AT) group type for an add, change, or transfer that contain a SURD01 record format; an opening Rental Notification (RN) group type; or, an Request for Authorization (RA) Change

ARMS Process Report

group type where the Date of Birth or the Rental Location value has been changed from a previous value. This called program is completed and removed from the job's program stack completely when *INLR is SETON.

@Files: (CRUD)

- AMSURTBL (-R--)
- BC026P (-R--)
- OFFDRB (-R--)
- AMSURD (CR-D)
- RAGBCHBT (-R--) * TEXAS GOVT SURCHARGE TO PRINT FOR 'I'nsurance

TYPE CUSTOMERS

@Constants:

'RECORD DELETED' is the message text for a value passed to AM1050V1 Add Maintenance Log Record program.

@Improvement Opportunities:

1.) If this program remained resident in the job's program stack, the overhead in opening all of the files could be reduced by making the file opens user-controlled to only occur if they are closed.

2.) Convert this OPM RPG/400 program to an ILE RPG module and have this data surcharge automatic generation logic executed as the transaction data set's record format's fields are being interfaced with the rental system database on the distributed rental application system host platform.

Process

Hierarchical numeric ID: 1.1.1.1.3.25

Coded name: AMRVTSQ

Name: PGM Assign Next Vendor Transaction Id (AMRVTSQ)

Comment: @Definition: This is the ARMS Assign Next Vendor Transaction ID OPM RPG Callable program.

@Purpose: To construct the next unique Vendor Transaction Identifier value for use in creating a new ARMS Rental Transaction Cross-reference File record.

@Operational Method:

Attempt to retrieve the LAST USED VENDOR TRANSACTION ID from the data area AMSEQCTL and lock it for update.

IF there was an error accessing this data area, load VENDOR TRANSACTION ID with 'ERROR' and return to the calling program.

IF there was no error in accessing and locking this data area, then increment its IDENTIFICATION NUMBER value by 1 and concatenate this 9 digit integer to the passed input 1 character MACHINE ID value and load the first 10 of 20 characters of a candidate VENDOR TRANSACTION ID.

Then, for every IDENTIFICATION NUMBER that is non-zero, ensure that an ARMS Rental Cross-Reference File record does not exist for the this derived Vendor Transaction ID by execution of the ARMS Cross-Reference Inquiry (AM2010V1) program in a loop until a return code of 'E' (No Cross-Reference Found) is sent, continuing to increment by 1 the numeric portion of this candidate VENDOR TRANSACTION ID.

ARMS Process Report

Update the data area with the last derived Vendor Transaction ID value when the RETURN CODE of 'E' was returned to this calling program along with the Last Changed Date, Time and Program ID.

Return the new Vendor Transaction ID to the calling program.

@Notes:

This program is executed by following ARMS never-ending programs (NEPs):

ARMS Database update and routing (AM0046V1)
ARMS Dispatch Outbound Transactions (AM0060V1)
ARMS Outbound transaction editor (AM0062V1) (For non-ARMS Rental Management Trading Partners' Electronic Invoicing.)

It is executed with the following parameters:

Input	1 character	MACHINE ID
Input	10 character	PROGRAM ID
Output	10 character	VENDOR TRANSACTION ID (to be used for cross-reference file record creation)

@Embedded Data/Constants:

'ERROR'
'E'
'AM2010V1'

@Improvement Opportunities:

Convert this OPM RPG/400 program to an ILE RPG service program.

Process

Hierarchical numeric ID: 1.1.1.1.3.26

Coded name: AM1099

Name: PGM Retrieve Next Transmission Control Id Number (AM1099)

Comment: @Purpose: To retrieve, generate and return the next ARMS internal Transmission Control ID number for use in transmissions from the distributed machine to the centralized host or to another distributed machine as in a transfer. This number is unique on its originating machine but is not unique across the whole ARMS.

@Operational Method:

The program is called passing two parameter fields, the first input and the second is output:

Input	CALLING PROGRAM ID
Output	TRANSMISSION CONTROL ID

Attempt to retrieve the next internal TRANSMISSION CONTROL ID value from the ARMS Next Outbound/Transfer Internal Transmission Control Sequence Number (AMRNAKSQ) file's single record and lock it for update.

If there is a database error, then load the output parameter for TRANSMISSION CONTROL ID NUMBER with the current timestamp's day of month, hour, minute and second (in the format: 0ddmmhhss, with a zero in the first leading position, followed by the 2-digit dd = day of month, followed by 6-digit current 24-hour time to the whole second.).

ARMS Process Report

If the record does not exist, then load the output parameter for TRANSMISSION CONTROL ID NUMBER with the numeric value of 1, increment that number by 1 and write a single AMRNAKSQ file record.

If the record does exist, then check if the retrieved NEXT TRANS SEQUENCE is zero, then load the output parameter for TRANSMISSION CONTROL ID NUMBER with the numeric value of 1. Else, load the output parameter for TRANSMISSION CONTROL ID NUMBER with the current NEXT TRANS SEQUENCE field value. Increment that NEXT TRANS SEQUENCE number field value by 1. If the updated NEXT TRANS SEQUENCE number field value is now zero, then set that value to 1. Update the single AMRNAKSQ file record.

Return the INTERNAL TRANSMISSION CONTROL ID TO BE USED to the calling program.

@Files: (CRUD)

- AMRNAKSQ (CRU-)

@Notes: When a transmission is to be sent to the trading partner, it is placed in the packaging file, AMPACK from where the packaging program, AM0120 picks it up and retrieves the next external transmission control id from a data area to assign it to this transmission. This external transmission control id exists only on the centralized host as opposed to the internal id that exists on all computer platforms and is unique only by machine.

@Improvement Opportunities:

- The file should be changed to a data area since there will always be only one record in the file, similar to AMSEQCTL (and use the empty file as a layout only for external data structure purposes). This may require an exclusive lock allocation (via an API or ILE RPG pointers) to wait less than 5 seconds, then reattempt until successful locking for input and writing the updated data storage, then issuing a deallocate/unlock. Also, this program should be duplicated to an RPG-ILE service program once some of the calling programs are converted to RPG-ILE programs. Additionally, CCARMSV1 no longer calls this program, therefore remove this dependent code. Lastly, this program is missing any program exception handling, like *PSSR subroutine calling AMPSSR. However, it may not be needed.

- 'CCARMSV1' used to check if is the value of the CALLING PROGRAM ID input parameter, then SETON *INLR since this program is being called by an interactive program. (CCARMSV1 no longer calls this program, this processing requirement was moved to CC00INV1, therefore remove this code.)

Process

Hierarchical numeric ID: 1.1.1.1.3.27

Coded name: AM1050V1

Name: PGM Create Sensitive Fields Change Log (AM1050V1)

Comment: @Purpose: To generate an ARMS Rental Transaction Database Files' Changes and Deletes (Maintenance) audit log file record with the passed input Vendor Transaction ID, File Prefix Code, Field Name suffix, the previous "before-change", its current "after-change" and the Functional Group Type values for ARMS database files' field value changes. This is used for auditing or troubleshooting purposes.

@Operational Method:

This program will accept a parameter list composed of most of the data fields in the AMMNTLOG record, excluding the Record Added Date and Time audit fields.

ARMS Process Report

(There are no record change stamp fields since this is a history of database changes or deletes log file.) The parameters (all input) are:

20 character Vendor Transaction ID
2 character File Code-File prefix
4 character Field Code-Field suffix
50 character Data - New value
50 character Data - Previous value
2 character Group Type Code
10 character Calling Program ID
5 character Employee ID

This program will accept input parameters provided and add a record to ARMS Rental Transaction Database Files' Changes and Deletes (Maintenance) audit log (AMMNTLOG) file. The record add stamp fields are loaded along with the passed input parameters values.

@Files:

AMMNTLOG (C---) Keyed by Vendor Transaction ID

@Improvement Opportunities:

- If any ARMS Database file fields increase in length beyond 50 characters, then the previous and current data value fields in AMMNTLOG and this parameter list (along with the programs that execute this program) will need to match this increase and the existing data will need to be converted to the changed format files. Likewise, if any of the ARMS Database file field names increase beyond 6 characters or the first 2 characters no longer denote file prefix, then these parameters, programs using this program and the AMFLTBL and AMFLDTBL fields will need to be changed also.

Process

Hierarchical numeric ID: 1.1.1.1.3.28

Coded name: AM1080V1

Name: PGM Generate Transaction Credits (AM1080V1)

Comment: @Purpose: To generate transaction credit log records and to accumulate the TOTAL CUMULATIVE REVENUE and TOTAL CUMULATIVE TRANSACTION COUNT for a trading partner.

@Operational Method:

1) Attempt to retrieve the associated Transaction Cross-Reference file, AMXREF, record by the passed Vendor Transaction ID to retrieve its Company Profile ID, Customer Transaction ID, Rental Location ID and Ticket ID. If unsuccessful retrieval, execute AM0097V1 program with passed Error Code equal to '98' and return to calling program.

2) Determine if an associated Transaction Volume Discount Profile (ARMSPR7) exists for this trading partner. If the associated profile record does not exist, then return to calling program.

3) If the last 6 characters of Location ID is blank, then the first 4 characters contain an ECARS Group/Branch ID. Therefore, use the Company Profile ID to attempt to retrieve the associated ARMSPR5 ARMS Company Profile ECARS Rental System Variables file record. Else, use the Company Profile ID to attempt to retrieve the associated ARMSPR6 ARMS Company Profile Claims Connection Rental System Variables file record. If an associated Company Profile file record was not

ARMS Process Report

retrieved, execute AM0097V1 program with passed Error Code equal to '67' and return to caller program.

4) If the passed Processing Code is 'P'ayment Advice, then perform detail calculations to compute the total amount billed to the trading partner.

5) If the passed Processing Code is 'P' or 'R', then perform detail calculations to compute the TRANSACTION CREDIT AMOUNT.

6.) Increment the retrieved ARMSPR5 or ARMSPR6 file record's PAYMENT ADVICE ACCUMULATED PAYMENT TOTAL COUNT (this is essentially the cumulative transaction count) and ACCUMULATED PAYMENT TOTAL AMOUNT (this is essentially the cumulative revenue).

8) Write record to AMTRNCR (Month-to-Date transaction credit file) after loading the following fields, including current date/time stamp fields:

- Populate the INSURANCE TOTAL RATE/DAY with the BILLED DAILY TOTAL AMOUNT (calculated above).

- Load the TRANSACTION CREDIT AMOUNT from the retrieved ARMSPR7 file record's DISCOUNT AMOUNT A field.

- IF the passed Processing Code is 'P', then load the TOTAL TICKET AMOUNT and the AMOUNT BILLED CUSTOMER from their associated fields from the retrieved AMIEBT file record's fields' values. ELSE (Processing Code = 'R' - Closing RN), clear the fields to zeros.

- IF the passed Processing Code is 'P', then load the INSURANCE TOTAL RATE/DAY field with the previously computed INSURANCE MINIMUM TOTAL AMOUNT value. ELSE (Processing Code = 'R' - Closing RN), clear the field to zeros.

- If the retrieved ARMSPR7 file record's DISCOUNT AMOUNT B field value is not zero, then do the following to calculate the ESTIMATED UTILIZATION FEE: If the retrieved ARMSPR7 record's DISCOUNT FREQUENCY B value is 'P' (Percent), then multiply the DISCOUNT AMOUNT B by the decimal ratio of the DISCOUNT FREQUENCY B field value. If the DISCOUNT FREQUENCY B value is 'L' (Lump) then use this value.

- Load the BILLED DATE from the retrieved AMIEBD record's ADD DATE.

- Load the following date subfields with the current system date: APPROVAL CENTURY/YEAR, APPROVAL MONTH, APPROVAL DAY.

@Complex Calculations:

BILLED DAILY TOTAL AMOUNT calculation:

1.) Attempt to retrieve the associated AMAUTD Authorization Detail file record for this Vendor Transaction ID to obtain the AUTHORIZED DAILY RATE, NUMBER OF DAYS AUTHORIZED, MAXIMUM DAILY RATE COVERED, POLICY MAXIMUM COVERED, BILL-TO PERCENT. If unsuccessful retrieval, execute AM0097V1 program with passed Error Code equal to '68' and return to caller program.

2.) If the NUMBER OF DAYS AUTHORIZED is greater than zero, then retrieve each associated AMIEBD Invoice Detail file record that has an ITEM CODE equal to 05 (Daily Rate Detail).

a.) For each record daily charge invoice detail record read,

i.) Save the least of ITEM RATE (Rental Rate Charged), AUTHORIZED DAILY RATE, MAXIMUM DAILY RATE COVERED as the RATE AMT. ii.) Save the lesser of ITEM QUANTITY (Days Charged) or NUMBER OF DAYS AUTHORIZED as the DAYS QUANTITY.

ARMS Process Report

iii.) If saved DAYS QUANTITY is greater than zero, then calculate
BILLED AMOUNT = RATE AMT * DAYS QUANTITY.

Subtract DAYS QUANTITY from NUMBER OF DAYS AUTHORIZED.
Accumulate BILLED DAILY TOTAL AMOUNT by BILLED AMOUNT.

c.) Multiply BILLED DAILY TOTAL AMOUNT by AUTHORIZED BILL-TO PERCENT
(AMAUTD) to give a new BILLED DAILY TOTAL AMOUNT value.

d.) If the POLICY MAXIMUM COVERED amount is not zero, compare it the
BILLED DAILY TOTAL AMOUNT. Add the lesser value to the appropriate retrieved ARMSPR5
/ ARMSPR6 file record's ACCUMULATED PAYMENT TOTAL AMOUNT value giving a new
ACCUMULATED PAYMENT TOTAL AMOUNT field value.

TRANSACTION CREDIT AMOUNT calculation.

1.) Retrieve (without update) the other rental system's Company Profile
File (ARMSPR5 or ARMSPR6) for this Profile ID. Add together the new ACCUMULATED
PAYMENT TOTAL COUNT fields from the two files (ARMSPR5 and ARMSPR6) with the
increment of 1 for this transaction, giving a TOTAL ACCUM. COUNT (If not found,
then set Error Code equal to '67').

2.) Use the current Company Profile ID and TOTAL ACCUM. COUNT result
from (1) to SETLL on ARMSPR7 and retrieve the next record to get the amount of the
transaction credit.

3.) If ECARS invoice, then retrieve the Group/Branch's associated Region
Code from the OFFDRB Office Directory Branch File associated Group/Branch record.
Else, it is a Claims Connection invoice, then leave region blank and use '7680' as
Group/Branch ID.

4.) Retrieve the associated AMIEBT Invoice Total File record to get the
Rental Ticket Total Amount and the Amount Billed to the Direct Bill Customer (error
code 68 if not found) .

@Notes:

This program is only executed as a called program from the AM0046V1
never-ending batch program in the ARMS subsystem only on the ARMS application
centralized host computer system platform whenever it processes the first closing
'RN' Rental Notification transaction data set for a non-zero Bill-To Percent
authorized rental whose profile indicates auto-generation of a (PM) (currently as
GEICO using ARMS/400), or, whenever it processes a 'PM' Payment Advice transaction
data set.

Executed with the following passed input parameters:

9/0 packed numeric	Transmission Control ID Number
5/0 packed numeric	Transmission Group Control ID Number
20 character	Vendor Transaction ID
1 character	Functional Group Transaction Type Processing

Code ('P'-PM or 'R'-Closing RN).

@Files: (CRUD)

ARMSPR7	(-R--)
AMAUTD	(-R--)
AMIEBD	(-R--)

ARMS Process Report

AMIEBT (-R--)
OFFDRB (-R--)
ARMSPR5 (-RU-)
ARMSPR6 (-RU-)
AMTRNCR (C--)

@Embedded Data/Constants:

'7680' is loaded as default non-ECARS location in the Group/Branch ID data structure field to load the Group ID and Branch ID fields in the written AMTRNCR file record.

PGMNAM and PPRPGM are both set to 'AM1080V1 '. (Suggest using the Program Data Structure subfield for Program ID in future.)

@Improvement Opportunities:

1.) Convert this OPM RPG program to an ILE RPG module that is part of the current AM0046V1 program.

2.) Since this program is attempting to do the same calculations that was done at Close a Rental Contract calculation time, I suggest that the computation of the Daily Rental Direct Billing Sub-Total Amount on each rental contract be stored in a new Rental Authorization Detail file's associated Sub-Total Amount for Daily Rental, along with Approved Surcharges Charged Sub-Total Amount and Taxes Charged Sub-Total Amount. Then, if the centralized ARMS database needs to retain this information for transaction credit purposes, a new internal format, IEBT02, can be loaded during "Generate Electronic Billing Invoice" in the transaction set and can be utilized and stripped off from the outbound transaction set that is sent to the trading partner. However, it would be simple if these new fields were interrogated from a centralized archive repository of closed rental contracts for the ARMS Transaction Credit Reports run monthly that primarily use the AMTRNCR file.

3.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.

4.) Delete the constant 'AM1080V1' and replace its usage with the Program Status Data Structure data element for PROGRAM ID.

5.) Convert program from OPM RPG to ILE RPG.

6.) Replace the use of the AM2010V1 program with the AM1010V1 program called for "I"nquiry.

Process

Hierarchical numeric ID: 1.1.1.1.3.29

Coded name: AM1010V1

Name: PGM Inquire/Update Cross - Reference (AM1010V1)

Comment: @Purpose: To retrieve for inquiry or update an existing associated ARMS Rental Transaction Cross-Reference File record when passed with the appropriate parameters.

@Operational Method:

- The following four input combinations can be passed to this program:
ARMS VENDOR TRANSACTION ID;
ARMS TRADING PARTNER COMP.PROFILE ID and CUSTOMER TRANSACTION ID;

ARMS Process Report

RENTAL LOCATION and the RENTAL CONTRACT/TICKET ID; or
RENTAL LOCATION and the RENTAL BRANCH RESERVATION ID.

Likewise, records may be updated or added to the ARMS Rental Cross-Reference File by executing this program in "Maintenance" mode. Only inquiry is allowed if executed in "Inquiry" mode.

1.) The RENTAL LOCATION is always required and either a non-blank TICKET ID or RESERVATION ID must be passed.

IF RENTAL LOCATION, TICKET ID and RESERVATION ID are blanks, then load the RETURN CODE with 'E' (Error).

2.) IF Rental Location and Reservation ID are passed with the program call, use AMXREFL2 to retrieve the requested ARMS Cross-Reference File record.

a.) IF Ticket number is also passed with the program call, retrieve the record using AMXREFL2. Then validate the Ticket number passed with the program call against the Ticket data field XATKNO.

b.) IF passed TICKET ID and retrieved TICKET ID are different, then load the RETURN CODE with 'T' (Ticket ID Mismatch) and return to the calling program.

c.) IF the passed TICKET ID is same as the TICKET ID found for the reservation in the ARMS Cross-Reference File, then update all fields in the parameter list from AMXREFL2, set the RETURN CODE to blanks (blanks indicate successful) and return to the calling program.

3.) IF non-blank RENTAL LOCATION and TICKET ID are passed with the program, use AMXREFL1 to retrieve the data.

IF no record found for RENTAL LOCATION and TICKET ID, then load the RETURN CODE with 'E' (Error).

4.) IF an ARMS COMPANY PROFILE ID is passed with the program call and a record is retrieved, compare this value with XACUID in AMXREFLn.

IF they are different, change the RETURN CODE to 'P' for 'Profile Error' and return to calling program.

5.) IF non-blank ARMS COMPANY PROFILE ID and CUSTOMER TRANSACTION ID are passed, retrieve record from AMXREFL3.

a.) IF TICKET ID / RESERVATION ID is passed, compare these with values retrieved from AMXREFL3.

IF they do not agree, change return code to 'T' / 'R'.

b.) IF RENTAL LOCATION is passed, compare this value with value retrieved from AMXREFL3.

IF they do not match, change return code to 'L'.

6.) IF an ARMS VENDOR TRANSACTION ID is passed, retrieve record from AMXREF.

a.) IF an ARMS COMPANY PROFILE ID is passed with the program call and a record is retrieved, compare this value with the associated field in the retrieved ARMS Cross-Reference File.

ARMS Process Report

IF they are different, change the return code to 'P' for 'Profile Error' and return to calling program.

b.) IF RENTAL LOCATION is passed, compare this value with the AMXREF value. IF they are different, change return code to 'L'.

c.) IF TICKET ID or RESERVATION ID is passed, compare these values with values retrieved from AMXREF.

IF they do not agree, change RETURN CODE to 'T' / 'R'.

7.) IF a record is retrieved from AMXREF or any of these logical files based upon the file use hierarchy, and additional data fields are passed, and more than one data mismatch occurs, change the return code to 'M' for multiple errors.

Hierarchy for record retrieval:

1.) IF ARMS VENDOR TRANSACTION ID is passed, and any other data fields, retrieve from AMXREF.

2.) IF ARMS COMPANY PROFILE ID and CUSTOMER TRANSACTION ID are passed, and any other data fields (except VENDOR TRANSACTION ID), retrieve from AMXREFL3.

3.) IF RENTAL LOCATION and RESERVATION ID are passed, retrieve from AMXREFL2.

4.) IF RENTAL LOCATION and TICKET ID are passed, retrieve from AMXREFL1.

RECORD MAINTENANCE FUNCTIONS:

1.) ACTION CODE passed is 'M' for maintenance. It is required to pass the VENDOR TRANSACTION ID for all maintenance transactions. Additional fields are optional, but only the physical file AMXREF is used for maintenance. ADD/CHANGE PROGRAM ID should also be sent.

2.) IF any existing AMXREF record is being maintained, the data fields are updated with the passed values along with record change stamp fields.

3.) IF new AMXREF record is being written, the data fields are updated with the passed values along with record add stamp fields.

4.) IF an AMXREF record is retrieved, compare the contents of each data field in the record with parameter containing that field.

IF they are different, create a maintenance log for the change. This is done only if parameter value is not blanks. Any non-blank value contained in the parameter list is logged and mapped to the AMXREF. AM1050V1 is called to create AMMNTLOG record.

RETURN CODE possible values:

blank: Successful retrieval / edit
L: Location passed is blank / invalid
#: Ticket/Reservation passed is blank
T: Ticket passed, unmatched on AMXREF record
P: Profile passed, unmatched on AMXREF record
E: No AMXREF record found
M: Multiple errors

ARMS Process Report

ACTION CODE possible values:

I: Inquiry
M: Maintenance

- IF the passed CALLING PROGRAM ID is NOT 'AM0046V1', then execute call to the ARMS Program Exception Handler 'AMPSSR' program, passing the CALLING PROGRAM ID and the Transaction's 96-character key value that is comprised of the passed TRANSMISSION CONTROL ID#, GROUP CONTROL ID#, GROUP TYPE CODE, CUSTOMER TRANSACTION ID and VENDOR TRANSACTION ID, padded with blanks.

@Notes:

This program is only executed by other ARMS never-ending programs that are processing a ARMS transaction request to inquire or update the ARMS Cross-Reference File.as being in error or not doable and rejected it from further processing. This program is currently called with parameters by the following programs:

ARMS Database Update and Routing program (AM0046V1)
ARMS Dispatch Rental Systems Request program (AM0060V1)
ARMS Distributed Edit and Dispatch Inbound Transactions program

(AM0061V1)

ARMS Distributed Edit of Generated Transaction Sets program (AM0062V1)

The parameters are internally defined in this program and programs executing this program are the following parameter fields:

Output	1 character	RETURN CODE
Input/Output	20 character	VENDOR TRANSACTION ID
Input/Output	5 character	TRADING PARTNER CO. PROFILE ID
Input/Output	20 character	CUSTOMER TRANSACTION ID
Input/Output	10 character	RENTAL LOCATION
Input/Output	6 character	RESERVATION ID
Input/Output	6 character	TICKET ID
Output	1 character	STATUS CODE
Output	8/0 numeric	STATUS CODE LAST CHANGED DATE
Output	6/0 numeric	STATUS CODE LAST CHANGED TIME
Output	10 character	STATUS CODE LAST CHANGED PGM ID
Output	1 character	MACHINE ID
Output	10 character	BRANCH CLAIMS OFFICE CODE
Output	1 character	RENTAL SYSTEM SOURCE ID
Output	8/0 numeric	RECORD ADD DATE
Output	6/0 numeric	RECORD ADD TIME
Output	10 character	RECORD ADD PROGRAM ID
Output	8/0 numeric	RECORD CHANGE DATE
Output	6/0 numeric	RECORD CHANGE TIME
Output	10 character	RECORD CHANGE PROGRAM ID
Input	2 character	GROUP TYPE CODE
Input	1 character	ACTION CODE

@Files: (CRUD)

- AMXREF (CRU-) AMXREFL1, L2, L3

@Embedded Data: (Constants)

'AM1010V1 ' is used as the program's name.

ARMS Process Report

The following values are named constants used for sending the ARMS Cross-Reference File (AMXREF) record's field name suffix for the passing of any field value modified as a FIELD CHANGED parameter field value in the call to the ARMS Transaction Database Maintenance Logging (AM1050V1) program.

'CUID'
'CUTI'
'RNLC'
'RENO'
'TKNO'
'STCD'
'STDT'
'STTM'
'STPG'
'MCID'
'SOID'
'TEID'
'ADDT'
'ADTM'
'ADPG'
'CGDT'
'CGTM'
'CGPG'

Also, there is a compile-time array to store the values for the formatted OS/400 commands:

```
CLOF OPNID(AMXREF )  
OPNDBF FILE(AMXREF ) OPTION(*ALL)
```

@Improvement Opportunities:

Converting this OPM RPG program to ILE RPG service program. It might negate the need for the OS/400 command execution OPNDBF and CLOF of the AMXREF file.

Replace the 'AM1010V1 ' constant with the PROGRAM ID field of the Program Status Data Structure.

Replace the internally defined parameter fields in this program and all programs executing this program with a database file with no record to be used as an externally defined data structure so that only one definition of the parameter's data elements exist, along with comments on their purpose and possible values.

This program can replace the call to the ARMS Cross-Reference File Inquiry (AM2010V1) program in many other programs if it is called in "Inquiry" mode.

Process

Hierarchical numeric ID: 1.1.1.1.3.30

Coded name: RTVMCHA

Name: PGM Retrieve Machine Emulation Attributes (RTVEMCHA)

Comment: @Purpose: To retrieve the emulated system (machine) platform attributes important for Enterprise programs that have been standardized for a machine.

This program is executed when executing the user-written command of the same name, "RTVEMCHA" (Retrieve Emulated Machine Attributes).

@Operational Method:

To avoid adversely impacting performance, all of these steps are executed only the first time a job executes the RTVEMCHA program. Subsequent execution in

ARMS Process Report

production will bypass program initialization steps, only returning previously loaded values.

Upon program initialization, do the following:

- executes the program to retrieve the system (machine) name (RTVSYSD)
- opens GACTL and MACHID files
- retrieves the system name's records from both GACTL and MACHID
- moves data from both GACTL and MACHID into its parameter list
- closes GACTL and MACHID files
- execute the /Q Check DEV's machine without emulation program (ERTVDEV)

to determine IF actually on "DEV". IF true, return the previously derived values to the calling program and end program and return to caller.

After program initialization has been performed and this inactive program was re-executed by the same job's calling program, only do the remaining processing step, return the previously derived values to the calling program.

@Files (CRUD)

MACHIDENPF (-R--) Machine Information File, via keyed access path
MACHID#1 by key element MACHINE ID (8 character)
GACTL (-R--) Global Attributes file, by key element MACHINE ID (8 character)

@Notes:

This program is a Internal Systems Support Group utility tool software callable program, executed with a single parameter defined in an external data structure in its physical file (\$#EMACH). All its output parameter subfield data elements, their attributes and a description can be found in source text member ELLIB/QDOCSRC/RTVEMCHA.

This program is especially important for each of our software applications needs the flexibility to accommodate the multi-national differences we will encounter in day-to-day business dealings since E-R-A-C is now operating outside the North American continent. For example, what we consider a standard method of displaying the date (MM/DD/YY) is not the standard in the United Kingdom; they expect to see it displayed as DD/MM/YY.

To facilitate expansion of business operations into the United Kingdom, and eventually into other European countries, a Global Attributes file (GACTL) was created to contain information that varies from country to country. By using the system name obtained through RTVSYSD, a program can access the GACTL file and use that information to tailor display screens and reports to its needs. For example, the GACTL file contains a field to tell programs to display the date as MM/DD/YY or DD/MM/YY.

Process

Hierarchical numeric ID: 1.1.1.1.3.31

Coded name: AM1085V1

Name: PGM Generate Underage Driver Surcharge (AM1085V1)

Comment: @Purpose: To delete any preexisting and write any needed ARMS Pre-Approved Authorized Surcharge Detail file (AMSURD) records with the CHARGE TYPE CODE = '20' (Underage Driver Surcharge) and the CHARGE ORIGIN CODE = 'U' (Underaged Driver Surcharge Generation program).

@Operational Method:

ARMS Process Report

1. Call the Retrieve ARMS Cross-Reference File Record (AM2010V1) program using the passed VENDOR TRANSACTION ID field value and its required parameters to retrieve the associated ARMS Profile ID value.

2. IF the returned SOURCE ID = 'B' (Branch Reservations), then call the Retrieve ARMS ECARS-Specific Profile File Record (AM2020V1) program to retrieve from the associated ARMS ECARS-Specific Profile File (ARMSPR5) record its flag field value for automatically generating underage driver approved surcharge authorizations (INSURANCE WILL PAY UNDERAGE='Y').

ELSE (SOURCE ID = 'C' - Claims Connection), call the Retrieve ARMS ECARS-Specific Profile File Record (AM2030V1) program to retrieve the same information from the associated ARMS-Claims Connection Specific Profile file (ARMSPR6) record.

3. IF this retrieved profile's flag field value is 'Y' (Yes), then do the following:

a. Using the passed VENDOR TRANSACTION ID, retrieve the associated Authorization Detail file (AMAUTD) record. IF successful, derive (compute) the DRIVER'S AGE by subtracting the DATE OF BIRTH from the RENTAL START DATE.

b. Call the Retrieve Rental Location's State Code Inquiry (AM2053V1) program, passing the called Retrieve ARMS Cross-Reference File Record (AM2010V1) program's returned parameter value for RENTAL LOCATION ID to retrieve the RENTAL LOCATION'S STATE CODE value.

c. Using the retrieved PROFILE ID value and RENTAL LOCATION'S STATE CODE value, retrieve the associated ARMS Profile Control Fields by State Code file (ARMSPR8) record.

IF unsuccessful, reattempt using the value '**' in place of the RENTAL LOCATION'S STATE CODE value. If still unsuccessful, execute return processing below.

d. IF the derived DRIVER'S AGE is greater than or equal to the the retrieved ARMS Profile Control Fields by State Code file (ARMSPR8) record's UNDERAGE THRESHOLD AGE value, then use the passed VENDOR TRANSACTION ID value and SURCHARGE TYPE CODE = '20' (Underage Driver) to retrieve and delete any associated AMSURD file record. For each record deleted, the program will write a record to the ARMS Maintenance Log file by the calling of Generate ARMS Maintenance Log Records (AM1050V1) program with loaded parameter values.

e. IF the derived DRIVER'S AGE is less than the the retrieved ARMS Profile Control Fields by State Code file (ARMSPR8) record's UNDERAGE THRESHOLD AGE value and the INSURANCE WILL PAY UNDERAGE is equal to 'Y', then write an associated ARMS Surcharge Detail file (AMSURD) record, using the passed VENDOR TRANSACTION ID value, SURCHARGE TYPE CODE = '20' (Underage Driver Surcharge Authorization Type Code), the retrieved ARMS Profile Control Fields by State Code file (ARMSPR8) record's UNDERAGE CHARGE AMOUNT value with a CHARGE FREQUENCY set to 'D' (Daily Charge) and the CHARGE ORIGIN CODE = 'U' (Underaged Driver Surcharge Generation program).

@Notes:

ARMSPR8 is currently empty.

This program is called from the AM0046V1 ARMS update never-ending program executing in a batch mode on the centralized ARMS application host computer system platform when processing a transaction data set for the following conditions: an Authorization (AT) group type for an add, change, or transfer that contain a AUTD01

ARMS Process Report

record format; an opening Rental Notification (RN) group type; or, an Request for Authorization (RA) Change group type where the Date of Birth or the Rental Location value has been changed from a previous value. This called program is completed and removed from the job's program stack completely when *INLR is SETON and RETURN is executed.

IMPORTANT: ARMS Profile Control Fields by State Code File (ARMSPR8) is currently empty in the production environment. Therefore, this program is called, loaded and initialized, then explicitly is ends, removed from the program stack and control is returned to the calling program.

@Files: (CRUD)

- AMAUTD (-R--) ARMS Authorization Detail File
- ARMSPR8 (-R--) ARMS Profile Control Fields by State Code File
- AMSURD (CR-D) ARMS Approved Surcharge Detail File

Process

Hierarchical numeric ID: 1.1.1.1.3.32

Coded name: AM0052

Name: PGM Determine Nearest Location by Phone Number (U.S. Only) (AM0052)

Comment: @Purpose: To determine the closest United States rental branch to a specified telephone number's area code and exchange using the Horizontal and Vertical Wire Coordinates locator database.

@Operational Method:

At the start, file VERHOR gives the vertical and horizontal coordinates of a specified area code and exchange number.

These coordinates are given in "units". There is 316 units per mile.

This program scans a 30 mile area around the VERHOR coordinates.

This is accomplished by calculating 4 new coordinates.

2 Horizontal and 2 Vertical.

The specified area code and exchange generates coordinates X2 and Y2 from the VERHOR file.

To "draw" the 30 mile box we subtract 9,480 units (30 miles) from X2 and Y2 to get X1 and Y1.

We then add 9,480 units to X2 and Y2 to get X3 and Y3.

Using the X1 and Y1 coordinate, attempt to retrieve the closest next record in the Daily Rental Offices (DROFF) file. This file contains the name, address...etc. of all daily vehicle rental office locations. Along with this information, the vertical and horizontal coordinates of each office are found in this file. By comparing these coordinates to the "box" coordinates we decide whether or not this office falls within the specified 30 mile area as read in as an input variable from the Search Input Parameters for Locator System (RANGE) data area.

The number of miles from the originally specified area code/exchange to each rental office is arrived at by calculating the square root of the combined horizontal and vertical coordinates of each rental office.

VWIRE = the vertical coordinate of the rental office.

HWIRE = the horizontal coordinate of the rental office

DVWIRE = the vertical coordinate of the original Area Code/Exchange

DHWIRE = the horizontal coordinate of the original Area Code/Exchange

ARMS Process Report

When a location is found that is within boundaries, check the "Prevent ARMS Reservations" flag. IF this flag is 'N', include the location. If this flag is 'Y' and there is no "Forward To Location" (DROAGB) field value, skip this location. IF this flag is 'Y' and there is a forwarding location, save the first forwarding location. IF no other Enterprise location is found, use the saved forwarding location.

All office information is stored in array "AC", which is sorted in ascending sequence according to mileage. The closest Enterprise office is used if one exists.

@Notes:

This program is currently executed only by the ARMS Detail Set Edits and Assign Location (AM0040V1) OPM RPG program called with the following parameters:

Input	3 character	AREA CODE	(comprised of non-leading zero integers only)
Input	3 character	EXCHANGE	(comprised of non-leading zero integers only)
Input	1 character	COMMISSION CODE	
Output	10 character	RENTAL LOCATION CODE	
Output	1 character	MACHINE ID	
Output	1 character	SOURCE ID	

@Files: (CRUD)

OFFDRGRPPF (-R--) OFFDG#2
VERHOR (-R--)
DROFF (-R--) DROFLF1, DROFFL2

@Embedded Data/Constants:

30 miles is the maximum radius/distance for closest Enterprise rental branch to pickup a renter.

@Improvement Opportunities:

Change the calling program to execute the AM0053 program instead for all trading partners but specifically profiled ARMS Trading Partner Company.

Process

Hierarchical numeric ID: 1.1.1.1.3.33

Coded name: AM0053

Name: PGM Determine Nearest Location by Longitude / Latitude using Phone Number or Postal Code (Canadian Only) (AM0053))

Comment: @Purpose: To locate offices in a specific area based on the telephone number provided and return the results and distance from point of inquiry to the requesting program. The request can be made for any Canadian Enterprise offices.

@Operational Method:

-This program has four main logic sections that:

1) Draws a variable size box around a centerpoint of a phone number. Program is passed phone number and a 30 mile box is drawn around this number. Program calls AT_002 and passes the

ARMS Process Report

phone# and size in the parm. AT_002 chains to the phone# file, gets the basepoint coordinates, and sends back via the parms the basepoint coordinates for that phone#, and the upper right and lower left longitude/latitude coordinates of the box, and a return code to indicate any errors. AT_002 adjusts and corrects the box size for the curvature of the earth and is accurate to within 400 ft on Ave.

2) Retrieves all the records from the branch office file that fall within the box.

3) Calculates the distance (in miles), from the phone number centerpoint, of all the records retrieved.

The program calls 'AT_001' program and passes a parameter that has the centerpoint (basepoint) and the longitude and latitude points of all records previously retrieved, and gets returned in the parameter the distances in miles for each set of points, and a PROGRAM RETURN CODE for errors. This is currently limited to 30 sets of points on each execution. IF more than 30 records are needed, a second(or third, fourth, etc.) execution can be made until all distances are calculated. This program scans a 30 mile area around the retrieved longitude and latitude coordinates and returns the .

4.) Find the rental location needed for this transaction. IF one does not exist, then send transaction to Claims Connection.

Make use of "Prevent ARMS Reservations" flag field (DROAMF) of the DROFF file. This field indicates whether unsolicited ARMS authorizations are to be sent to a group/branch. IF this field is 'Y', then send the authorization to the forwarding group/branch (if not blank) from the DROFF record. IF this flag is 'N', include the location. IF this flag is 'Y' and there is no "Forward To Location" field (DROAGB) value, skip this location. IF this flag is 'Y' and there is a forwarding location, save the first forwarding location. IF no other Enterprise location is found, use the saved forwarding location.

@Notes:

This program is currently executed only by the ARMS Detail Set Edits and Assign Location (AM0040V1) OPM RPG program called with the following parameters:

Input	10 character	TELEPHONE NUMBER
Output	10 character	RENTAL LOCATION CODE
Output	1 character	MACHINE ID
Output	1 character	SOURCE ID

It executes the following programs:

AT_002 - Draw box around a centerpoint.

AT_001 - Calculate Distance between points.

** Will notify the application department when problem occurs executing programs.

@Files: (CRUD)

OFFDRGRPPF (-R--) Office Directories Group Master file, accessed by the OFFDG#2 access path key element, GROUP ID (2 character)
DROFF (-R--) Daily Vehicle Rental Office Locations file, accessed by the DROFFL7 access path key element, LONGITUDE and LATITUDE (both 9/6 numeric)

@Embedded Data/Constants:

ARMS Process Report

30 miles is the maximum radius/distance for nearest Enterprise rental branch to pickup a renter.

'7680' is the default rental location for Claims Connection.

'C' is the default MACHINE ID host platform (System ID='CENTRAL') and SOURCE ID (Claims Connection) values.

'ARMS' is the constant TERMINAL ID value for input parameter to the execution of the 'AT_001' program.

@Improvement Opportunities:

Change the calling program (AM0040V1) to execute this program for all location once all rental locations have been loaded into the Longitude and Latitude Locator by Telephone Number database for all trading partners but specifically profiled ARMS Trading Partner Companies.

Process

Hierarchical numeric ID: 1.1.1.1.3.34

Coded name: NUS017A

Name: PGM Determine Nearest Enterprise Location by Phone or Postal Code (NUS017A)

Comment: @Purpose: To locate 1 to 9 offices in a specific area based on the telephone number or postal code provided and return the results and distance from point of inquiry to the requesting program. The request can be made for any offices from the DROFF (NR : Daily Rental & Non-Enterprise Office Loc's) file, only Enterprise offices, or only Enterprise airport offices.

@Operational Method: The following parameters must be passed to the program with appropriate values set based on task at hand.

Input Parm:

Location Phone#
Location Postal Code
Open Only Flag
Maximum Allowable Distance
Airport Branches Only Flag
Department ID
Accepting Reservations Flag
Retrn Forward-to Locatn Flag
Requested Date/Time
Time Zone Group/Branch ID
Opening Time Allowance
Closing Time Allowance
Return Code
Branches to Return 1-9

Output Parm:

Location ID Array - these are the locations that satisfy the input search criterion.

Mileage Array - these are the corresponding distances for the locations from the reference point.

@Notes:

1) A testing stub called LOCTRUT exists in library ELARMS to test the program.

2) In order to receive only Enterprise offices, the calling program must do the following overrides :

ARMS Process Report

Overriden file	Override to file
Droffl2	Droffl3
Droffl7	Droffl9

In order to exclude Ford locations, the calling program must do the following overrides.

Overriden file	Override to file
Droffl2	Droffl4
Droffl7	Droffl10

@Files	(CRUD)
DROFF	(-R--)

Process

Hierarchical numeric ID: 1.1.1.1.3.35

Coded name: AM2020V1

Name: PGM Retrieve ARMSPR5 Information (AM2020V1)

Comment: @Purpose: To access ARMSPR5 (ECARS Profile) and ARMSPR1 (Trading Partner General Profile) record and return the details to the calling program.

@Operational Method:

Executed with the following entry parameters and their usage:

Output 1 alpha	Program Return Code
Input 5 alpha	Trading Partner Profile ID
Output 10 alpha	ECARS Closing Calculations to Compute the
Direct Bill-To Total Due	
Output 11,2 numeric	Payment Total Amount
Output 1 alpha	Underaged Drivers Allowed (Y/N) indication
Flag	
Output 1 alpha	Insurance pays underage driver indication
Flag	
Output 1 alpha	Trading Partner has capability to recieve
Electronic Billing indication flag	

@Files: (CRUD)

ARMSPR1 (-R--)

ARMSPR5 (-R--)

@Improvement Opportunity:

This program and AM2030V1 could be combined into a single program, especially if the ARMS Trading Partner Profile database files are further normalized.

Process

Hierarchical numeric ID: 1.1.1.1.3.36

Coded name: AM2030V1

Name: PGM Retrieve ARMSPR5 Information (AM2030V1)

ARMS Process Report

Comment: @Purpose: To Access ARMSPR6 (Claims Connection Profile) and ARMSPR1 (Trading Partner General Profile) record and return the details to the calling program.

@Operational Method:

Executed with the following entry parameters and their usage:

Output	1 alpha	Program Return Code
Input	5 alpha	Trading Partner Profile ID
Output	10 alpha	Claims Connection Closing Calculations to
Compute the Direct Bill-To Total Due		
Output	11,2 numeric	Payment Total Amount
Output	1 alpha	Underaged Drivers Allowed (Y/N) indication
flag		
Output	1 alpha	Insurance pays underage driver indication
flag		
Output	1 alpha	Trading Partner has capability to recieve
Electronic Billing indication flag		

@Files: (CRUD)

ARMSPR1 (-R--)
ARMSPR6 (-R--)

@Improvement Opportunity:

This program and AM2020V1 could be combined into a single program, especially if the ARMS Trading Partner Profile database files are further normalized.

Process

Hierarchical numeric ID: 1.1.1.1.3.37

Coded name: RAS013A

Name: PGM Determine Nearest Enterprise Location by Phone or Postal Code (RAS013A)

Comment: @Purpose: To locate 1 to 9 offices in a specific area based on the telephone number or postal code provided and return the results and distance from point of inquiry to the requesting program. The request can be made for any offices from the DROFF (NR : Daily Rental & Non-Enterprise Office Loc's) file, only Enterprise offices, or only Enterprise airport offices.

@Operational Method: The following parameters must be passed to the program with appropriate values set based on task at hand.

Input Params:

Location Phone#
Location Postal Code
Country Code
Open Only Flag
Maximum Allowable Distance
Airport Branches Only Flag
Department ID
Accepting Reservations Flag
Retrn Forward-to Locatn Flag
Requested Date/Time
Time Zone Group/Branch ID
Opening Time Allowance

ARMS Process Report

Closing Time Allowance
Return Code
Branches to Return 1-9

Output Parm:

Location ID Array - these are the locations that satisfy the input search criterion.

Mileage Array - these are the corresponding distances for the locations from the reference point.

@Notes:

1) A testing stub called LOCTRUT exists in library ELARMS to test the program.

2) In order to receive only Enterprise offices, the calling program must do the following overrides :

Overriden file	Override to file
Droffl7	Droffl9

In order to exclude Ford locations, the calling program must do the following overrides.

Overriden file	Override to file
Droffl7	Droffl10

@Files	(CRUD)
DROFF (-R--)	

Process

Hierarchical numeric ID: 1.1.1.1.3.38

Coded name: NUS018A

Name: PGM Verify Branch is Open (NUS018A)

Comment: @Purpose: To send a GPBR and have the open/closed status returned. The request time can be the current date and time or a time in the future. The program uses the three Nat Res policy files, NRALPA2, NRPCDG, and NRGBM. It also uses GPBRTZ and a time zone reference field must be sent.

@Operational Method: The following paramters can be passed to the program:

Requested Group/Branch ID
Requested Date/Time
Time Zone Group/Branch ID
Opening Time Allowance
Closing Time Allowance
Within Opening Time Allowance
Within Closing Time Allowance
Open Flag
Return Code

The Open Flag and Return Code are the only return fields.

The program checks the Nat Res Policy Files to determine if the requested branch is open or closed.

Closing Time Allowance is available but not currently used by ARMS.

ARMS Process Report

@Files (CRUD)

NRALPA2 (-R-)
NRPCDG (-R-)
NRGBM (-R-)

Process

Hierarchical numeric ID: 1.1.1.1.3.39
Coded name: DQAM55V1
Name: DTQ Input for Program AM0055V1 (DQAM55V1)
Comment: @Definition: DQAM55V1 is a data queue used to provide input to PGM AM0055V1 which maintains the files used for ARMS Online Reporting, accessed via ARMS/400 (green screen or gui).

Process

Hierarchical numeric ID: 1.1.1.1.3.40
Coded name: AM0055V1
Name: PGM Online Reporting File Write/Update Program (AM0055V1)
Comment: @Purpose: To update and write records to the ARMS Group Type Summary File AM096P which keeps track of the number of times a group/type is transmitted for a customer profile and its vender transaction ID. Also to update and write records to the ARMS Online Reporting Detail file AM095P which keeps details on a rental keyed on customer profile and vender tran ID. To log transactions to ARMS Current days transaction for Sync program AM097P.

@Operational Method:

- Wait for entry(s) to exist in DTAQ (DQAM55V1).
- When a shutdown request is received (group type - SD), end program.
- When a non-shutdown request is received:
 - Create a record in AM096P01 to keep track of the number of transactions for a vendor tran id. If a record already exists add 1 to the total and update.
 - Check to see if a record exists in any of the following files by keying on Vender Tran ID: AMAUTD, AMRPRD, AMRNTD, AMSURD, AMIEBT, AMADJD, AMXREF
 - If a matching record is found move all of the appropriate data from AMXREF, AMAUTD, AMRPRD, AMADJD, AMRNTD, AMSURD, AMIEBT to the ARMS Online Reporting Detail File AM095P.
 - If there is not an existing record in AM095P00 one will be created, otherwise the existing record will be updated. All previous data in an existing record will be replaced with all data found in the files.
- Check to see if a record exists in AM097P01. If record exists, skip record. If record is not found, write a record to the file.

@Files: (CRUD)

- AMXREF (-R--)
- AMADJD (-R--)
- AMAUTD (-R--)

ARMS Process Report

- AMIEBT (-R--)
- AMRNTD (-R--)
- AMRPRD (-RU-)
- AMSURD (CR-D)
- AM0095P00 (CRU-)
- AM009601 (CRU-)
- AM009701 (CR--)

@Embedded Data/Constants:

'YES'
'NO '

Process

Hierarchical numeric ID: 1.1.1.1.4

Coded name:

Name: AUT Send Transaction for Distributive Processing (AM100-AM101)

Comment: @Definition: The automatic process of sending the transaction from the centralized machine to the distributed machine for processing, via ICF files.

Process

Hierarchical numeric ID: 1.1.1.1.4.1

Coded name: AM0100

Name: PGM Distribute Routed Transaction from Centralized Host System (AM0100)

Comment: @Purpose: To send a received (inbound) or transferred ARMS transaction set from the centralized host platform to the distributed host platform.

@Operational Method:

Execution is started with a single MACHINE ID (1 character) input parameter field value.

For program initialization:

- Retrieve the current platform's CURRENT SYSTEM (MACHINE) NAME by execution of the Retrieve System Name (RTVSYN) program.

- Concatenate the Recieve Routed Ditributed Transaction on Distributed Host System Program Name, 'AM0101', with the passed input MACHINE ID value for deriving the specific Program Device Name.

- IF CURRENT SYSTEM NAME = 'DEV', change the ICF workstation device file EVOKE program's library to 'ELARMS', ELSE, make the EVOKE program's library 'ELLIB'

- Attempt to acquire this Program Device. Attempt 5 times before signalling an error. Introduce a delay of one minute between each attempt.

- IF the Program Device is acquired successfully, Evoke the derived ARMS Distribution Incoming Receive Program.

After program initialization has completed, for a repetitive cycle until a shutdown data queue entry is received, the following is performed:

ARMS Process Report

- Receive the next keyed data queue entry from the specific keyed data queue for the key equal to the passed input MACHINE ID parameter value. Immediately from the DQAMAPH data queue until empty, then infinitely wait for DQAMAP1 data queue entries.

- Attempt to retrieve an associated record from the AMAPPS file using the data queue entry as the retrieval key value.

- IF at least one associated AMAPPS file record exists, then do the following:

- - Send the key to the ARMS Distribution Incoming Receive program.
- - Read all ARMS Set file records with the ARMS Application Data Queue Key.
- - Send each data record to the ARMS Distribution Incoming Receive program. When all records are read and sent, send a blank record.

@Files: (CRUD)

- AMAPPS (-R--)

@Embedded Data/Constants:

'AM0101' is the evoked program's specified name.

'ELARMS2' and '*LIBL' for the evoked program's specified possible library names.

'DEV' to determine if being executed on the development (not production) software environment platform.

'*DOWN' and 'SD' as the Profile ID and Group Type Code to indicate a data queue entry of a shutdown request.

'DQAMAPH' and 'DQAMAP1' are the Hold Distribution Input and the Transactions to Distribute Input data queue names.

'DLYJOB DLY(60)' is the OS/400 command executed to cause 1 minute delays between the 5 Attach attempts.

@Improvement Opportunities:

Convert this OPM RPG program to ILE RPG program.

Possibly change this communication method from ICF to CPI-C.

Process

Hierarchical numeric ID: 1.1.1.1.4.2

Coded name: AM0101

Name: PGM Receive Routed Distributed Transaction on Distributed Host System (AM0101)

Comment: @Purpose: To receive routed transaction set from the centralized host platform.

@Operational Method:

- Acquire the program device 'AM0101'
- Receive routed transaction input from the centralized ARMS host platform.
- IF data was received
 - Write records to the distributed Application Transaction File (AMAPP)
 - Send data queue entry to Edit/Update/Distribute Transaction (DQAM61)

ELSE

ARMS Process Report

- Send data queue entry to Edit/Update, Distribute Transaction (DQAM61)
- Release Device.

ENDIF

@Files: (CRUD)

- AMAPP (C---)

@Notes: The centralized ARMS host platform EVOKES the receive job on the distributed rental system host platform using an ICF connection. Presently, the distributed rental system host platform, excludes BIRELAND, VGERMANY, and UK.

Process

Hierarchical numeric ID: 1.1.1.1.4.6

Coded name: RTVSYS

Name: PGM Retrieve System Name (RTVSYS)

Comment: @Purpose: To retrieve the system name to pass back to another calling program.

@Operational Method:

Executed with the single output parameter, RETRIEVED SYSTEM NAME (8 Character).

Retrieve the current system's network attribute of system name.

IF the RETRIEVED SYSTEM NAME = 'DEV', then execute the ISS DEV Machine Emulation (DEVMCHE) OPM CL program.

Process

Hierarchical numeric ID: 1.1.1.1.5

Coded name:

Name: AUT Interface with Rental Systems (AT/CN/EX/CM)

Comment: @Definition: The automatic process of dispatching a data queue entry for an appropriate program to update the Rental database be it either ECARS or CLAIMS CONNECTION.

Process

Hierarchical numeric ID: 1.1.1.1.5.1

Coded name:

Name: AUT Interface with ECARS (AT/CN/EX/CM)

Comment: @Definition: The automatic process by which the ECARS Rental database files are updated due to a request received from the Rental Management Trading Partner (AT, CN, EX, CM).

Process

Hierarchical numeric ID: 1.1.1.1.5.1.1

Coded name: EC00ATV1

Name: PGM Interface EC with Rental Authorization (EC00ATV1)

Comment: @Purpose: This is an ARMS-to-ECARS database update interface program that processes Rental Direct Billing Authorization 'AT' Functional Group Type for

ARMS Process Report

Adds or Changes sent by ARMS insurance Customers or automatically generated authorization changes. This program will also process previously authorized reservation Rental Location's Group-to-Group 'AT' transfers.

@Operational Method:

This NEP receives the 'ATB' keyed input data queue entries from the input data queue (DQAM61V1) that program AM0061V1 generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQANDST data queue and end this program.

For any non-shutdown data queue entry, read all of the associated ARMS Application Interface Input Transaction file (AMAPP) records.

IF the flag for Centralized Adjuster Phone number is 'Y' then retrieve from CUSTMAST the office phone number from that file record based upon customer number and use this phone number to populate ECARS data files, else the adjuster phone number supplied in the ADJD01 format will be used. This flag is retrieved via a 'B1' call to AM2090V1.

IF any COMD01 or SURD01 record formats were found, execute the Update Electronic Messages program (ECMSGV1) with 3 input parameters, the APPD01 record format's Reservation ID (6 character), the data queue entry value (96 character), and the CALLING PROGRAM ID (10 character) to write the electronic messages to ECARS message database.

IF while attempting to retrieve any existing records that are locked for update, the transaction is deferred by sending the data queue entry to a secondary input data queue (DQAM6BV1) to be reedited by the AM0061V1 program for later processing and a deferral notice data queue entry is sent to the primary input data queue (DQAM61V1) to notify that program that a deferred transaction exists (both data queue entries sent with key = 'DIS'). This will trigger the program AM0061V1 to start the wait timer so that when the wait is elapsed it can receive from the secondary data queue.

IF an authorization transfer is received, then a new ARMS authorized branch reservation is created, along with any applicable instructions, callback and prewrite information, such as date of birth.

IF an authorization add is received that has an invalid or no open rental contract/ticket or reservation identifier in it, then a new authorized rental reservation is created, along with any applicable instructions and prewrite information, such as date of birth.

IF an authorization add or change is received that has a valid open rental contract (ticket) and a reservation identifier in it, then update both the existing unauthorized open rental ticket and its "linked" branch reservation with the authorization information.

IF an authorization add is received that has a valid open rental contract (ticket) but no reservation identifier in it, then update the existing open rental ticket with the authorization information, a new authorized rental reservation is created, and this reservation is "linked" to the open contract. However, IF the open rental contract was opened using another non-ARMS authorized branch reservation or national reservation, then break the "link" of the previous open rental contract identifying information from the previous reservation. When a NatRes link is broken, a record is created in the file, NRDBTRAN, to appropriately credit NatRes.

ARMS Process Report

In all cases that an existing branch reservation is updated with an authorization for the first time or an authorized branch reservation is created, a 'DIS' keyed data queue entry is sent to the DQAM60V1 data queue as input to the Dispatch Rental Systems Request (AM0060V1) program. This should cause an authorization confirmation (AC) group type transaction data sent to be dispatched for generation and sent.

In all cases that an existing open rental contract is updated with an authorization add or change, the associated adjuster callback direct billing authorization information is updated. For an authorization add only, the EXTENSION DATE is recomputed by adding this input transaction's data set's authorization detail (AUTD01) record format's AUTHORIZED DAYS to the open rental contract's START CHARGES DATE.

IF, however, the resulting EXTENSION DATE is a Saturday or Sunday, check the Trading Partner Profile's WEEKEND SLIDE FLAG. ..IF the WEEKEND SLIDE METHOD CODE is "Y"es and the EXTENSION DATE is Saturday or Sunday, then "slide" (change) the EXTENSION DATE to the date for the following Monday.

..IF the WEEKEND SLIDE METHOD CODE is "M"odified and the EXTENSION DATE is Sunday, then "slide" the EXTENSION DATE to the date for the following Monday.

Likewise for an authorization add only, callback detail notes are generated to indicate the recomputed EXTENSION with an "Authorized Until mm/dd/yy" and "By: AdjusterLastName*AdjusterFirstName*" formatted text where the "mm/dd/yy" is the computed new extension date and the Adjuster's name is derived from the data provided in the Adjuster Detail (ADJD01) record format.

@Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters.

This program was changed to not remove the *TX REMB miscellaneous government tax unconditionally on open contracts, but to instead check a flag in the Valid Business Types that the special TEXAS GOVERNMENT SURCHARGE is applied to (RAGBCHBT) file (via logical access path RAGBBT, by Business Type, Group ID and Branch ID) to see whether or not this tax should be charged.

@Files: (CRUD)

File	AMAPP	(-R--)	ARMS Application Interface Input Transaction
	RACPENDC	(-R--)	Open ECARS Rental Contracts/Tickets in progress of being changed or closed control file
	RACBRNUM	(CRU-)	Branch Reservation ID Control File
	RACBRMST	(CRU-)	Branch Reservation Master File
	RACBRWRT	(CRU-)	Branch Reservation Prewrite File
	RACINS	(CRU-)	Branch Reservation/Open Rental Contracts Insurance
Information File	RACBRIST	(CRU-)	Branch Reservation Special Instruction
	ECSURCHG	(CR-D)	ECARS Authorized for Rental Direct Billing Surcharges
File	RACMAST	(-RU-)	Open Rental Contract/Ticket File
	RACSMAS	(CRU-)	ECARS Open Rental Contract/Ticket Supplemental
Information File	CB007P	(CRU-)	Rental Callback Control File, accessed by RACCBCTL logical view, by Group/Branch ID and Document ID
	RACCBDET	(CR--)	Rental Callback Detail Notes File
	RACCK	(-RU-)	Branch Reservation Credit Check File
	CUSTMAST	(-R--)	Customer Master File

ARMS Process Report

RACCBIOC (-) Rental Consolidated Callback Control File
RACCBK3 (CRUD) Rental Consolidated Callback Control File
RACCB3 (CR-D) Rental Callback Consolidation Skeleton File - Shop
OFFDRB (-R--) Office Directories Internal Organization Branch

Master File

NR025P (-RU-) National Reservations Master File, accessed by
NR025L00 logical view, by National Reservation ID
NRDBTRAN (CRU-) National Reservations Database Transactions Update File

@Embedded Data/Constants:

'Until' used to provide a preformatted Callback Detail Notes phrase.

'/' used to as a date separator for date output into a Callback Detail

Notes text.

'ARMS CUST-' used to provide a preformatted RACBRIST File Sequence 001
record's comment text.

Process/shutdown codes:

'AT'

'SD'

Format IDs:

'APPD01'

'ADJD01'

'ADJD02'

'AUTD01'

'COMD01'

'INSD01'

'RNTD01'

'RNTD02'

'RPRD01'

'SURD01'

'RATD01'

Comments:

'VEHICLE NEEDED IMMEDIATELY'

'UNIT NEEDED AT OTHER#'

'UNIT NEEDED AT HOME#'

'UNIT NEEDED AT OFFICE#'

'INS CO WILL PAY'

'/DY' used as a Branch Reservation Rate Quoted Comment for the quoted
daily rental rate.

'Non numeric data received in Transmission/Group Control ID'

'LEC00ATV1' is used as the JOB NAME.

'EC00ATV1' used as the CALLING PROGRAM ID.

'ATB' as the input key used for receiving keyed data queue entries.

Constants for formatting numeric fields to text:

'/DAY'

'/MAX'

'.'

'0123456789' all valid numeric characters to use for validation of all
numeric character data element fields.

'RATE + TAX'

' + TAX'

'RATE +SCHG'

' +SCHG'

ARMS Process Report

The following OS/400 Override Printer file commands are executed through execution of the 'QCMDEXC' program to change the output queue names by insertion of the Group/Branch ID (4 character) in the output queue name to cause the printing of an open rental contract/ticket or the executed Print a Branch Reservation to print at the associated rental location's office plain paper printer:

```
OVRPRTF FILE(QPRINTP ) OUTQ(P      P) FORMTYPE(PLAINPAPER)
DLTOVR FILE(QPRINTP)
```

The following OS/400 database file override commands are executed through execution of the 'QCMDEXC' program to decrease the locked record/object wait times for update files from the object default of 1 minute to 5 seconds:

```
OVRDBF FILE(RACBRNUM) WAITRCD(5)
OVRDBF FILE(RACBRMST) WAITRCD(5)
OVRDBF FILE(RACBRWRT) WAITRCD(5)
OVRDBF FILE(RACINS)   WAITRCD(5)
OVRDBF FILE(RACBRIST) WAITRCD(5)
OVRDBF FILE(ECSURCHG) WAITRCD(5)
OVRDBF FILE(RACMAST)  WAITRCD(5)
OVRDBF FILE(RACCBCTL) WAITRCD(5)
OVRDBF FILE(RACCBDET) WAITRCD(5)
OVRDBF FILE(RACSMAST) WAITRCD(5)
OVRDBF FILE(RACCCK)   WAITRCD(5)
OVRDBF FILE(RACCBK3)  WAITRCD(5)
OVRDBF FILE(RACCB3)   WAITRCD(5)
OVRDBF FILE(NR025L00) WAITRCD(5)
OVRDBF FILE(NRDBTRAN) WAITRCD(5)
```

@Improvement Opportunities:

- 1.) Convert this OPM RPG program to an ILE RPG program.
- 2.) Delete the 'EC00ATV1' program name and 'LEC00ATV1' job name constants and specify/use the Program Status Data Structure's PROGRAM ID and JOB NAME data elements for these purposes.
- 3.) Move the "Authorized Until mm/dd/yy" and "By: AdjusterLast*AdjusterFirst*" formatted text comment templates into an external ARMS Table File that contains the preformatted callback detail note automated comments.
- 4.) Add a new executable program to check for an associated existing record in the file RACPENDC (CRU-) "Open ECARS Rental Contracts/Tickets in progress of being changed or closed control file" for checking if OK to process and if so, output a record until the operation is complete, then delete the RACPENDC record that was written to clear way for other transactions.

Process

Hierarchical numeric ID: 1.1.1.1.5.1.2
Coded name: EC00EXV1
Name: PGM Interface EC with Rental Extension/Termination of Authorization (EC00EXV1)
Comment: @Purpose:

To update the rental database with the authorized extension days or termination date received from the trading partner.

@Operational Method:

ARMS Process Report

This program endlessly receives the 'EXB' keyed input data queue entries from the input data queue (DQAM61V1) that program AM0061V1 generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQANDST data queue and end this program.

For any non-shutdown data queue entry, read all of the associated ARMS Application Interface Input Transaction file (AMAPP) records.

IF the any COMD01 or SURD01 record formats were found, execute the Update Electronic Messages program (ECMSGV1) with 3 input parameters, the APPD01 record format's Reservation ID (6 character), the data queue entry value (96 character), and the CALLING PROGRAM ID (10 character) to write the electronic messages to ECARS message database.

IF while attempting to retrieve any existing records that are locked for update, the transaction is deferred by sending the data queue entry to a secondary input data queue (DQAM6BV1) to be reedited by the AM0061V1 program for later processing and a deferral notice data queue entry is sent to the primary input data queue (DQAM61V1) to notify that program that a deferred transaction exists (both data queue entries sent with key = 'DIS').

IF the received transaction set's STATUS CODE is not "E"xtend nor "T"erminate, then reject the transaction set back to the trading partner's adjuster as a "data domain error".

In all cases that an existing open rental contract is updated with an authorization extension or termination, the associated open contract file record's and its adjuster callback file record's direct billing authorization information is updated if the adjuster information is changed.

IF this transaction is an authorization termination (CRED01:CRESTS='T'), then update the callback's current EXTENSION DATE is replaced with this input transaction's data set's customer request for extension detail (CRED01) record format's non-zero TERMINATION DATE, regardless if the new value of the EXTENSION DATE is a Saturday or Sunday date. Likewise, a callback detail note record is generated to indicate the replaced EXTENSION DATE with an " Rental Terminated On: mm/dd/yy" and "By: AdjusterLastName*AdjusterFirstName*" formatted text where the "mm/dd/yy" is the replaced EXTENSION DATE and the Adjuster's name is derived from the data provided in the Adjuster Detail (ADJD01) record format. Update the number of authorized days in the Reservation record.

IF this is an authorization extension (CRED01:CRESTS='T'), with DAYS TO EXTEND is zero, then the current EXTENSION DATE remains. This will allow ARMS Trading Partner Insurance Company to send a termination request. (The second phase will allow ARMS Trading Partner Insurance Company to receive "RE" and send "EX" transactions.) Likewise, a callback detail note record is generated to indicate the same value EXTENSION DATE with an " Rental Terminated On: mm/dd/yy" and "By: AdjusterLastName*AdjusterFirstName*" formatted text where the "mm/dd/yy" is the current value of the callback's EXTENSION DATE and the Adjuster's name is derived from the data provided in the Adjuster Detail (ADJD01) record format. Update the number of authorized days in the Reservation record.

IF this is an authorization non-zero extension, then the non-zero EXTENSION DATE is recomputed by adding this input transaction's data set's customer request for extension detail (CRED01) record format's DAYS TO EXTEND to the open rental contract's current EXTENSION DATE. .

ARMS Process Report

Likewise for an authorization extension for a non-zero DAYS TO EXTEND only, a callback detail note record is generated to indicate the recomputed EXTENSION with an "Rental Extended Until: mm/dd/yy" and "By: AdjusterLastName*AdjusterFirstName*" formatted text where the "mm/dd/yy" is the computed new extension date and the Adjuster's name is derived from the data provided in the Adjuster Detail (ADJD01) record format.

Update the number of authorized days in the Reservation record.

@Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters.

@Files: (CRUD)

File	AMAPP	(-R--)	ARMS Application Interface Input Transaction
	ECEXTCTL	(CRU-)	ECARS Pending Extension Request Control File
	RACMAST	(-RU-)	Open Rental Contract/Ticket File
	RACCBDET	(C---)	Rental Callback Detail Notes File
	RACBCOC	(-R--)	Rental Consolidated Callback Control File
	RACCBK3	(CRU-)	Rental Consolidated Callback Control File
	CB007P00	(-RU-)	Open Rental Contracts Callbacks Control File
	CB032P00	(-RU-)	RMS Callback Contracts by Group/Branch/Ticket
	RACBRMST	(-RU-)	ECR Branch Reservations Master File

@Embedded Data/Constants:

Detail Notes	'Rental Extended Until:'	used to provide a preformatted Callback phrase.
Detail Notes	'Rental Terminated On:'	used to provide a preformatted Callback phrase.
Detail Notes	'Rental Extended 0 days'	used to provide a preformatted Callback phrase.
	'DQAM61V1 '	
	'EC00EXV1 '	
	'OVRDBF CB007P00 WAITRCD(*IMMED)'	
	'OVRDBF RACMAST WAITRCD(*IMMED)'	
	'OVRDBF RACMAST WAITRCD(*IMMED)'	

'/' used to as a date separator for date output into a Callback Detail Notes text.

Logicals:

'Y'
'N'
'0'
'1'

Process/shutdown codes:

'EX'
'SD'

Format IDs:

'APPD01'
'ADJD01'
'CRED01'
'COMD01'

Comments:

ARMS Process Report

'LEC00EXV1' is used as the JOB NAME.
'EC00EXV1' used as the CALLING PROGRAM ID.
'EXB' as the input key used for receiving keyed data queue entries.
'0123456789' as all valid numeric characters to use for validation of all numeric character data element fields.

The following OS/400 Override Printer file commands are executed through execution of the 'QCMDEXC' program to change the output queue names by insertion of the Group/Branch ID (4 character) in the output queue name to cause the printing of an open rental contract/ticket or the executed Print a Branch Reservation to print at the associated rental location's office plain paper printer:

```
OVRPRTF FILE(QPRINTP ) OUTQ(P    P) FORMTYPE(PLAINPAPER)
DLTOVR FILE(QPRINTP)
```

@Improvement Opportunities:

1.) Delete the 'EC00EXV1' program name and 'LEC00EXV1' job name constants and specify/use the Program Status Data Structure's PROGRAM ID and JOB NAME data elements for these purposes.

2.) Change the program's use of the database files used for update purposes before opened for update via the following OS/400 database file override commands are executed through execution of the 'QCMDEXC' program. This would be to decrease the locked record/object wait times for update files from the object default of 1 minute to 5 seconds.

3.) Add a new executable program to check for an associated existing record in the file RACPENDC (CRU-) "Open ECARS Rental Contracts/Tickets in progress of being changed or closed control file" for checking if OK to process and if so, output a record until the operation is complete, then delete the RACPENDC record that was written to clear way for other transactions.

4.) Move the "Rental Extended Until:", "Rental Terminated On:", "Rental Extended 0 days" and "By: AdjusterLastName*AdjusterFirstName*" formatted text comment templates into an external ARMS Table File that contains the preformatted callback detail note automated comments.

5.) Prior to opening the user-controlled open files that are used to READ or CHAIN for UPDATE, execute an "Override Data Base File" command to change the record-lock wait times from the default 60 seconds to 5 seconds as was done in the EC00ATV1 and EC00CNV1 programs.

Process

Hierarchical numeric ID: 1.1.1.1.5.1.3
Coded name: EC00CNV1
Name: PGM Interface EC with Rental Cancellation / Denial of Authorization (EC00CNV1)
Comment: @Purpose:

To update the rental database with the cancellation information received from the trading partner to either cancel a previously ARMS-authorized open contract and/or open branch reservation, or, to deny billing responsibility in response to a request for authorization on a branch reservation or an open rental contract (ticket).

@Operational Method:

ARMS Process Report

This program endlessly receives the 'CNB' key input data queue entries from the input data queue (DQAM61V1) that program AM0061V1 generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQANDST data queue and end this program.

For any non-shutdown data queue entry, read all of the associated ARMS Application Interface Input Transaction file (AMAPP) records.

IF there are no deferred transactions while processing, then execute the Update Electronic Messages program (ECMSGV1) with 3 input parameters, the APPD01 record format's Reservation ID (6 character), the data queue entry value (96 character), and the current CALLING PROGRAM ID (10 character) to write the electronic customer messages to ECARS message database.

IF while attempting to retrieve for update any existing file records that are locked for update, then the transaction is deferred by sending the data queue entry with key = 'DIS' to a secondary input data queue (DQAM6BV1) and a deferral notice data queue entry with key = 'DIS' to a primary input data queue (DQAM61V1) to be reedited by the AM0061V1 program for later processing.

IF the received transaction set's reservation is not found, the reservation is closed, or closed ticket that had been opened using the ARMS-authorized reservation, then error the transaction and page ARMS On-call.

IF the received transaction set's reservation is closed, then execute the Internal/External Error Handling Program (AM0098) to Reject the transaction.

IF the received transaction set's retrieved unauthorized reservation has been voided, discontinue any further processing of this transaction.

IF the received transaction set's retrieved rental contract has been voided, then the transaction is deferred by sending the data queue entry with key = 'DIS' to a secondary input data queue (DQAM6BV1) and a deferral notice data queue entry with key = 'DIS' to a primary data queue (DQAM61V1) to be reedited by the AM0061V1 program for later processing. AM0061V1 will then handle the Ticket record not being found appropriately.

IF the received transactions set's retrieved reservation is authorized, then do the following:

..IF the received transaction set's reservation is found, is ARMS authorized, and, a ticket is currently being opened using the reservation (BRTKT = 999999), then defer this transaction as stated previously.

..IF the received transaction set's open ticket and/or reservation is found, is/are ARMS authorized, and available, then reset its direct billing indication (leave the Bill-To ID and related information), load the RATE QUOTED COMMENT with "ARMS Direct Bill Canceled By AdjusterLastName,AdjusterFirstName, and clear the ARMS authorization information from Branch Reservation File (RACBRMST) and from Ticket Master Files (RACMAST & RACSMST) file records and its associated Callback Control and update. Create a Callback Control for the Reservation if it does not exist. Generate a Callback Detail Note file record with the comment "ARMS Direct Bill Canceled By AdjusterLastName,AdjusterFirstName" in NOTE #1, along with the associated adjuster's claim center internal CUSTOMER NAME from the Customer Master File (CUSTMAST) as "At customername" in NOTE #2. Update or create the consolidated callback control files. Callbacks for a reservation attached to an Incomplete Ticket will be created or updated using the Reservation number.

ARMS Process Report

..IF the received transaction set's reservation is found, is NOT ARMS authorized, and is available for update (regardless if linked to open contract/ticket or not), then reset its (and its associated open rental contract/ticket) direct billing indication (leave the Bill-To ID and related information), load the reservation's RATE QUOTED COMMENT with "ARMS Direct Bill Denied By AdjusterLastName,AdjusterFirstName", in the Branch Reservation File (RACBRMST) and the Ticket Master File (RACMAST) file records and its associated open rental contract/ticket file and update. Update the callback and the consolidated callback control files, using the open rental contract/ticket identifier if it is completely open (has a non-zero CONTRACT DATE), else use the reservation's identifier. Create a Callback Control for the Reservation if it does not exist. Generate a Callback Detail Note file record with the comment "ARMS Direct Bill Denied By AdjusterLastName,AdjusterFirstName" in NOTE #1, along with the associated adjuster's claim center internal CUSTOMER NAME from the Customer Master File (CUSTMAST) as "At customername" in NOTE #2.

@Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters.

@Files: (CRUD)

AMAPP (-R--)
RACBRMST (-RU-)
CB007P00 (CRU-)
RACCBDET (C---)
RACMAST (-RU-)
RACCBCOC (-R--)
RACCBCTK (C-U-) ECR Callback Consolidation Skeleton File,
by Group ID, Branch ID, Rental Contract/Document ID via the RACCBK3
logical access path.
RACBCBS (C-U-) ECR Callback Consolidation Skeleton File - Shop,
by Group ID, Branch ID, Rental Contract/Document ID via the RACCB3
logical access path.
CUSTMAST (-R--)
RACSMAS (-RU-)
EMSG (C---)

@Embedded Data/Constants:

'ARMS'
'ARMS Direct Bill Denied By'
'ARMS Direct Bill Canceled By'

@Improvement Opportunities:

Process

Hierarchical numeric ID: 1.1.1.1.5.1.4
Coded name: EC00CMV1
Name: PGM Interface EC with Customer Message (EC00CMV1)
Comment: @Purpose:

To update the rental database with any text messages received from the trading partner in the functional group type 'CM'.

@Operational Method:

ARMS Process Report

This program endlessly receives the 'CMB' keyed input data queue entries from the input data queue (DQAM61V1) that program AM0061V1 generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQANDST data queue and end this program.

For any non-shutdown data queue entry, read all of the associated ARMS Application Interface Input Transaction file (AMAPP) records.

IF any COMD01 or SURD01 record formats were found, execute the Update Electronic Messages program (ECMSGV1) with 3 input parameters, the APPD01 record format's Reservation ID (6 character), the data queue entry value (96 character), and the CALLING PROGRAM ID (10 character) to write the electronic messages to ECARS message database.

@Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters.

@Files: (CRUD)

AMAPP (-R--)
RACBRMST (-R--)
RACMAST (-R--)

@Embedded Data/Constants:

'EC00CMV1' used as the CALLING PROGRAM ID.
'CMB' as the input key used for receiving keyed data queue entries.

@Improvement Opportunities:

- 1.) Convert this OPM RPG program to an ILE RPG program.
- 2.) Delete the 'EC00CMV1' program constant and specify/use the Program Status Data Structure's PROGRAM ID data element for this purpose.

Process

Hierarchical numeric ID: 1.1.1.1.5.1.6
Coded name: AM2090V1
Name: PGM Retrieve ARMS Data (AM2090V1)
Comment: @Purpose:

To retrieve specified information from the ARMS database.

@Operational Method:

This program is executed using a single 256-byte data structure parameter for both input and output (see @Notes for a detailed description).

- Determine which type of information is being requested using the passed input parameter identification code.

-- Passed input parameter's data element for Data Structure ID Code specifies the associated information retrieval processing description:

ARMS Process Report

'B1' - Retrieves AMXBCO file record trading partner's branch claims office (BCO) information by the passed ENTERPRISE CUSTOMER ID

'B2' - Retrieves AMCLSTBL file ARMS Vehicle Class Description and Rate Code Table information by the passed ECARS VEHICLE CLASS CODE

'B3' - Retrieves AMXREF Rental Transaction Cross-Reference, AMXBCOL1 Branch Claims Office*, and ARMSPR1 Trading Partner attributes information by the passed ENTERPRISE CUSTOMER ID, RENTAL LOCATION ID and RESERVATION//TICKET ID

'B4' - Retrieves AMXREF, AMXBCOL1*, ARMSPR1, along with the Rental System Application Specific** (ARMSPR5 or ARMSPR6) information by the passed RENTAL LOCATION ID and RESERVATION//TICKET ID

'B5' - Retrieves AMXREF, ARMSPR1, along with the Rental System Application Specific** (ARMSPR5 or ARMSPR6) information by the passed RENTAL LOCATION ID and RESERVATION//TICKET ID

'B6' - Retrieves AMXREF, ARMSPR1 information by the passed RENTAL LOCATION ID and RESERVATION//TICKET ID

'B7' - Retrieves AMXBCOL1, ARMSPR1 file information by the passed ENTERPRISE CUSTOMER ID

'B8' - Retrieves ARMSPR1 file record information by the passed ENTERPRISE CUSTOMER ID

'B9' - Retrieves AMXREF file record information by the passed RENTAL LOCATION ID and RESERVATION//TICKET ID

(*Note: Branch Claims Office information is retrieved only when the retrieved Rental Transaction Cross-Reference indicates that it is not currently authorized or the Identification Code is 'B4'. This condition was needed since Trading Partner Profile ID was stored in two places. Undesirable results occurred prior to this condition.)

(**Note: Rental Application System Profiles consists of two files: One for ECARS used by ERAC branches for the rental administration of their group's vehicles; and one for Claims Connection used by the National Reservations Call Center staff for the administration of third-party non-ERAC vehicle rentals. The Source ID on the Rental Transaction Cross-Reference ('B' = Branch ECARS; 'C' = Claims Connection) determines which file the information is retrieved.

- IF the passed Identification Code does not match those accepted by this program, generate a program exception error and return an unsuccessful Completion Status Code ('D') to the calling program.

- IF the requested information record(s) are successfully retrieved, return the values from those file(s) record(s) along with a successful Completion Status Code ('A') to the calling program.

- IF any record was not found, return blank value(s) from the specific file record along with a successful Completion Status Code ('A') to the calling program.

- IF a database error occurred, then generate a program exception error and return an unsuccessful Completion Status Code ('D') to the calling program.

- When the passed Identification Code is 'B4', determine if the trading partner's branch claims office is currently active for electronic billing.

ARMS Process Report

@Notes:

This program is currently executed with a single 256 character string input/output parameter that is further redefined using the empty ANDSB#V1 file as an external data structure. The external definition contains two important input fields. The first is the Data Structure Identification Code, which is always a two position alpha field in the first two positions of the data structure. The second immediately follows the first and is a one position Program Return Code. A value of 'A' returned in this position to the calling program indicates the processing request was completed successfully. The total maximum length is 256 positions, and the unused space has been defined as a "filler" (future expansion) field.

The data structure data elements descriptions are:

Input	2 character	ARMS Identification Code
Output	1 character	Program Return Code
Input	7 character	Enterprise Customer ID
Input	10 character	Rental Location ID
Input	6 character	Reservation ID
Input	6 character	Ticket ID
Input	2 character	ECARS Vehicle Class Code
Input	8,0 numeric	Transaction Date (in CCYYMMDD format)
Input	10 character	Calling Program ID
Output	20 character	Vendor Transaction ID
Output	20 character	Customer Transaction ID
Output	5 character	ARMS Profile ID
Output	1 character	ARMS Rental Transaction Cross-
Reference File Record Status Code		
Output	1 character	Machine ID
Output	1 character	Source ID - Branch Reservations /
Claims Connection		
Output	10 character	Branch Claims Office - AMXBCO
Output	14 character	Enterprise Car Class Code
Output	20 character	ARMS Car Class Description
Output	10 character	Billing Program
Output	1 character	Generate Extensions from Rental -
P1FG10		
Output	1 character	Generate Batch Extensions from
Rental - P1FG11		
Output	1 character	Electronic Bill(Y/N) - P1YN12
Output	1 character	Generate Policy Max Note P1YN17
Output	1 character	View Ticket/Reservation 1 Day Early
- P1YN21		
Output	1 character	Only Days Needed For New Callback
Request - P1YN22		
Output	1 character	Electronic Messaging - P1FG24
Output	1 character	Allow Cust Tran Id on Fax Documents
- P1YN25		
Output	1 character	Allow Underaged Driver - P1YN26
Output	1 character	Send ELCD01 Format on RA - P1YN27
Output	1 character	Generate Electronic & Paper Bills -
XBYNBL		
Output	1 character	Insurance Pays Underage Surcharge
Output	1 character	Weekend Slide Y=Sat/Sun M=Sun Only
N=No Slide - P1		
Output	1 character	R=RA I=IN B=Both N=No
Output	1 character	How to Populate RATD01?
Output	1 character	ARMS Special Condition
Output	1 character	Send Adjuster Name on RA Y/N

ARMS Process Report

Output	1 character	Central and Adjuster Phone #
Output	1 character	Expansion Flag Field
Output	1 character	Expansion Flag Field
Output	1 character	Expansion Flag Field
Output	1 character	Expansion Flag Field
Output	1 character	Expansion Flag Field
Output	1 character	Expansion Flag Field
Output	1 character	Expansion Flag Field
Output	1 character	Expansion Flag Field
Output	1 character	Expansion Flag Field
Output	1 character	RC Cancel Assignment Surcharge
Handling		
Output	73 character	Filler (Future Expansion)

NOTE: New functions desired by this program have to be closely coordinated with the rental company application (ECARS or Claims Connection) requesting the new function desired in this program. All objects involved with the new function must be moved into production at the same time.

IF this program is executing on a development platform, terminate the program when returning to the calling program for deallocation of resources among many testers within an ARMS testing environment.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer). Therefore, ARMS will receive a data queue entry to generate an electronic invoice (and create an associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) when the trading partner never sent to ERAC (via ARMS) any initial rental authorization.

@Files: (CRUD)

AMXBCO (-R--) ARMS Branch Claims Office Cross-reference File, using access path AMXBCOL1 by Enterprise Customer ID.

AMXREF (-R--) ARMS Cross-reference File, accessed by Vendor Transaction ID

Accessed by Rental Location ID, Ticket ID

accessed by Rental Location ID, Reservation ID

ARMSPR1 (-R--) ARMS Profile File - Application Specific Flags
ARMSPR5 (-R--) ARMS Profile File - ECARS Application Specific

Data

ARMSPR6 (-R--) ARMS Profile File - CLAIMS Application Specific

Flags

AMCLSTBL (-R--) ARMS Vehicle Class/Rate code file

@Embedded Data/Constants:

'AM2090V1' as the PROGRAM ID
'B1' through 'B9' are the ARMS Identification Code for the type of information retrieval requested.

@Improvement Opportunities

1.) Convert from OPM RPG program to ILE RPG service program.

2.) Delete the 'AM2090V1' constant and use the Program Status Data Structure's left-adjusted PROGRAM ID value for the same purposes.

3.) Replace the execution of the current ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.

Process

Hierarchical numeric ID: 1.1.1.1.5.1.8
 Coded name: MILMATH
 Name: PGM Manipulate Ambiguous Dates (MILMATH)
 Comment: @Purpose:

To compare, add or subtract any passed input dates parameter field values. This program handles everything that the 'CCDATE' program does, with new standards, plus a module/program to handle addition/subtraction and comparison of ambiguous dates in different formats.

@Operational Method:

This program accomplishes the same thing as any of the RPG operation codes it replaces (ADD, SUBTRACT, etc., see FUNCTIONS) except that it makes century part of the equation. Basically, any dates calculated through MILMATH are expanded to include Century first and the fields in the program remain unchanged. Century is determined for all Standard Dates by comparing the year portion to 40. Century is determined for all Date Of Birth Dates by assuming no Birth dates can be in the future and no one is older than 100.

FUNCTIONS:

ADDITION: Of numbers representing dates to numbers representing dates, and of numbers to numbers representing dates.

SUBTRACTION: Same as Addition only MILMATH also will handle subtracting a number representing Date Of Birth from a number representing a standard date.

COMPARISON: Between numbers representing dates. Using COMP opcode.

EXPANSION: Of numbers representing dates with YY to YYYY.

Is currently executed with the following parameters:

Action: Input 1 Character - flag for the type of operation that MILMATH is to perform. (see FUNCTIONS.)

(YY is YYYY or YYYYMMDD)

depending on Length 2,4 or 6.)

- (A) = YY + # of Years
- (B) = YY + YY
- (C) = YY COMPARE to YY
- (I) = YY Expand to YYYY
- (S) = YY SUBTRACT # of Years
- (T) = YY SUBTRACT YY
- (U) = YY SUBTRACT Date Of Birth

Length: Input 2,0 Numeric - This parameter describes the length of data being sent in DATE1 and DATE2 described below.

(2) = YY format- 2 digits.

digits.

(6) = YYMMDD format- 6 digits.

Date1: I/O 8,0 Numeric - This parm receives factor 1 of a calculation to be done and sends it back with century added.

Sent:	Returned:
YY	CCYY
YYMM	CCYYMM
YYMMDD	CCYYMMDD

Date2: I/O 8,0 Numeric - This parm receives factor 2 of a calculation to be done and sends it back with century added.

Sent:	Returned:
YY	CCYY
YYMM	CCYYMM
YYMMDD	CCYYMMDD

Result: Output 8,0 Numeric - This parm returns the result of the calculation if applicable.

HI Indic: Output 1 Character - This parm returns the HI indicator result of all applicable calculations.

LO Indic: Output 1 Character - This parm returns the LO indicator result of all applicable calculations.

EQ Indic: Output 1 Character - This parm returns the EQ indicator result of all applicable calculations.

@Embedded Data/Constants:

Possible valid ACTION CODE values:

'A' = Add Number of Years
 'B' = Add Years
 'C' = Compare Years
 'I' = IF Compared Years
 'S' = Subtract Number of Years
 'T' = Subtract Year from Year
 'U' = Subtract Date Of Birth Year from Current Year to compute age

@Improvement Opportunities:

Convert the programs that execute this program to ILE RPG and do all date manipulation by converting the values to the same format Date data types and performing like data type operations on the dates.

Process

Hierarchical numeric ID: 1.1.1.1.5.1.9

Coded name: CCRAZX

Name: PGM Retrieve GPBR Tax Detail / Cvt Sys Time to Br Time (CCRAZX)

Comment: @Purpose:

To retrieve a rental branch tax accounting information and convert the system time and date (currently United States Central Time Zone for all computer

ARMS Process Report

platforms) to the rental branch's local time and date by using the TIME ZONE ADJUSTMENT field value (that is +n or -n hours from the Central Time Zone time).

@Operational Method:

This program is only executed when called by another program using two parameter fields: Group/Branch Tax Data for 215 characters, and Time Zone Conversion Data for 41 characters.

The Time Zone Conversion Data 41 characters are further redefined as:

1 alpha	Input Action Code	(possible values: A B G R T)
4 alpha	Input Group/Branch ID	
6 alpha	Input Time (hhmmss - where "ss" is optional)	
8 alpha	Input Date (mmddccyy)	
1 alpha	Input Time Format Flag	(possible values: M - Military/23-hour or S - Standard/12-hour with AM/PM)
6 alpha	Branch Military Time (hhmmss)	
6 alpha	Branch Standard Time (hhmmAM)	
6 alpha	Branch Modified Date (mmddyy)	
2 alpha	Century (cc)	
1 alpha	Program Return/Error Code	(possible values: B G R T blank)

Input Action Code dictates what this program is to return to the calling program. The codes are listed as follows:

A = Sent Group ID and Branch ID. Retrieve associated Group/Branch Tax file information and convert/adjust passed time/date according to time flag.

B = Sent Group ID and Branch ID. Only check if that Group/Branch ID value exists in Group/Branch Tax file. If not, load 'B' to the PROGRAM RETURN/ERROR CODE parameter subfield.

G = Sent Group ID only. Check if that Group ID value exists in Group/Branch Tax file. If not, load 'G' to the PROGRAM RETURN/ERROR CODE parameter subfield.

R = Sent Group/Branch ID. Retrieve associated Group/Branch Tax file record's information. Do not convert/adjust time/date. If not, load 'R' to the PROGRAM RETURN/ERROR CODE parameter subfield.

T = Time conversion only. (Not used by any program currently calling CCRAZX.)

Based on the outcome of the input codes and the data provided to the program, one of the following return codes will be generated:

(blank) = No error
B = Group/Branch ID record not found in the ECARS Group/Branch Tax File.
G = Group ID not found.
R = Group/Branch ID record not found for Group/Branch Tax file record information retrieval request.
T = Standard Input time invalid or cannot be interpreted because none of the characters "AaPp" (for AM/PM or am/pm) are in the passed TIME field's position 5. Impossible since this program does not edit for time errors. It assumes time sent for conversion is correct.

ARMS Process Report

Retrieves the associated GPBRTZ Group/Branch Time Zone Adjustment file record that indicates the time zone difference (in hours and fractions of hours) from United States Central time.

Program CCRAZX will expect the following implied questions from the calling programs:

"What is the branch time/date equivalent to the system time/date?"

"What are the individual government charge computation requirements for this branch?"

No editing occurs in this program - otherwise calling program is responsible for validity of input data.

A record for the branch code must exist in the BC026P file. The value of the time zone adjustment field TZADJ must be -5 to less than 19.

@Files: (CRUD) (---- signifies the file usage is only as an external data structure.)

- BC026P00 (-R--)
- GPBRTZ (-R--)
- DTPARM (----)

@Constant / Embedded Data:

AaPp - CHK array to check for a valid AM(am)/PM(pm) indication value.

@Improvement Opportunity: If the internal data structures were defined externally, then any changes to the structure would be prone to less errors and only a recompile of all programs that utilized them would need to be performed to implement the change. (NOTE: The data structure to copy into each program is in ELLIB/QDOCSRC/CCRAZX. However, some comments are out of date because they are no longer valid. This includes the specifications in the ARMS NEPs that check if the PROGRAM RETURN/ERROR CODE = "A", but no code in the CCRAZX program will load this value to that parameter subfield.)

There are several problems with the "time zone adjustment" field BC026P:TZADJ. While these pose no problem under the current range of time zones covered by existing branches, there are basic conceptual errors which show up when the data is applied on a global basis. The first problem is the existing range, -12 to +12, specified for the field. This is a set of 25 values to be applied to 24 time zones; taking St Louis/Central time adjustment as 0, the -12 and +12 values overlap on the (global) 18th zone. Because the above set of values does not take into account the International Date Line, its use returns date errors. For systems which use Central Standard (global time zone 6) as the system time, the correct value is the offset from St Louis time; -5 to less than 19. However, the UK machine uses Greenwich as its system time (global time zone 12). In this case values from -12 to less than +12 are correct. The meaning of the field and its values needs to be given a single definition. This has been accommodated by the two fields in the newer GPBRTZ file.

GPBRTZ File Description: This file holds the time zone information for each Enterprise Rental group/branch. Time Zone information includes:

- a.) Time zone offset from the International Date Line (was Greenwich Mean Time - GMT or UTC - Universal Coordinated Time) used to determine in which time zone a branch exists. (e.g., Eastern, Central, Hawaii).
- b.) Time zone adjustment with respect to the Central Time used by the Two O'Clock programs and to print time on the tickets.
- c.) Day Light Savings Observe indicator to indicate if branch observes Daylight Savings time period (from first Sunday in April through last Sunday in October.)

ARMS Process Report

@Notes:

Programs EC00ATV1, EC00CNV1, EC00EXV1, ECMSGV1, AM2051V1 and module CBZ004A use this program to adjust the St. Louis (System) time value to the local group/branch's time for the transaction data set being processed. EC00ATV1 also uses this program during a different execution to retrieve the Sales Tax/Surcharge Account ID value to use for the decision of which formatted text should be used in the Branch Reservation file record's MAXIMUM AMOUNT text field.

Process

Hierarchical numeric ID: 1.1.1.1.5.1.10
Coded name: ECMSGV1
Name: PGM Update Electronic Message (ECMSGV1)
Comment: @Purpose:

To update ECARS Electronic Message database files with the received messages from the trading partner.

@Operational Method:

This program is called from ECARS NEP's that run in the ARMS system that process COMD01 or CBKD01 formats. It uses the transaction's passed key value received from the DQAM61V1 input data queue to access AMAPP file data and load the APPD01, ADJD01, COMD01, CBKD01 and SURD01 proprietary EDI record formats' data into the appropriate external data structures used within this program. This information is then used to update an existing or write a new ECARS Message Control File (RACMSC) record and write corresponding new Message Detail (RACMSD) file records. No data is returned to the calling program.

@Notes:

This program is executed with the following 3 input parameters:

6 character	Reservation ID
96 character	AMAPP Application Interface Input Transaction File key
10 character	Calling Program ID

This program does not Set On *INLR so that it can be reexecuted without reperforming the program retrieval, memory loading/addressing and program initialization.

@Files: (CRUD)

AMAPP	(-R--)	Distributed APPLICATION INTERFACE input transactions file
RACBRMST	(-R--)	Branch Reservations Master File
RACMAST	(-R--)	ECARS Open Rental Contracts/Tickets Master File
RACBCOC	(-R--)	ECARS Rental Callbacks Consolidation Control File
RACMSC	(CRU-)	Electronic Message Control File
RACMSD	(C---)	Electronic Message Detail File

@Embedded Data/Constants:

Actual Constants:

' - Sent by'
'Sent by'
'Surcharge Authorized: '
'/Day'

ARMS Process Report

'/Rental
' Percent

Embedded Data:

'EC00CMV1'
'EC00ATV1'
'AT' (Authorization Maintenance proprietary EDI transaction group type
code)
'T' (Authorization Transfer proprietary EDI transaction set type code)
'APPD01' (Internal ARMS routing header proprietary EDI record format)
'ADJD01' (Adjuster Detail proprietary EDI record format)
'CBKD01' (Callback Detail proprietary EDI record format)
'COMD01' (Comment proprietary EDI record format)
'SURD01' (Surcharge Approved for Direct Billing proprietary EDI record
format)
'R' (Rental Document ID Reservation Type Prefix)
'D' (Rental Document ID Rental Contract/Ticket Type Prefix)
'*' (LastName*FirstName* delimiter)
'ARMS' (Record Created By Employee)
'D' (Per Rental-Daily Amount Surcharge)
'L' (Per Rental-Lump Amount Surcharge)
'P' (Per Rental-Percentage Surcharge)
'+' (Surcharge Amount Positive Sign)
'-' (Surcharge Amount Negative Sign)
'05' (Government Surcharge Type Code)
'I'
'E'
'Y' (Yes)
'N' (No)
'T'
'A'
'M' ('CCRAZX' program execution input parameter for Action Code)

@Improvement Opportunities:

Convert this OPM RPG program to an ILE RPG service program comprised of
ILE RPG modules.

Process

Hierarchical numeric ID: 1.1.1.1.5.1.11
Coded name: CCRS10
Name: PGM Print Reservation Detail (CCRS10)
Comment: @Purpose:

To notify and alert the rental branch that a rental authorization has
been received from the rental management trading partner. Furthermore, display
specific information about the reservation: Renter's name; address; phone number;
Insurance replacement information and any special requests (Vehicle-Van, requires
pickup date/time).

@Operational Method

- Retrieve branch reservation detail record
- IF reservation was created thru ARMS
Print fields and headings. Calculate renter pay percent
- Convert flag fields to constants (pickup status, loss type)
- Build surcharge print lines

ARMS Process Report

- Retrieve last 10 messages ONLY
- Override the printer file to the Group/Branch output queue. Change
formtype to PLAINPAPER

@Files: (CRUD)

RACBRMST (-R--)
RACPAYOP (-R--)
LIFEM#1 (-R--)
ECSURCHG (-R--)
RACMSG1 (-R--)

Process

Hierarchical numeric ID: 1.1.1.1.5.2

Coded name:

Name: AUT Interface with Claims Connection (AT/CN/EX)

Comment: @Definition: The automatic process by which the Claims Connection
database files are updated as a result of a request sent by the Rental Management
Trading Partner (AT, CN, EX,).

Process

Hierarchical numeric ID: 1.1.1.1.5.2.1

Coded name: CC00ATV1

Name: PGM Interface CC with Rental Authorization (CC00ATV1)

Comment: @Purpose:

To update the claims connection database with any authorization
information received from the trading partner.

@Operational Method:

- Wait indefinitely for the next DQAM61V1 data queue entry with a key of
ATC'.
- IF the receive data queue entry is a shutdown request, send that
request to data queue DQANDST to inform the distributed send program it has ended.
- IF the receive data queue entry is a non-shutdown request THEN
- Open programs files, if not previously opened.
- Read the transaction data set records from AMAPP and
and populate the associated record format.
- IF authorization is a change request and there's no ticket update
the reservation.
- IF authorization is a change request and there is a ticket, update
the ticket. Communicate this change to ARMS by
forwarding a data queue entry to DQAM60V1.
- IF authorization is an add or transfer request and there's no
ticket, create the reservation. Communicate this change to
ARMS by forwarding a data queue entry to DQAM60V1.
- IF authorization is an add or transfer request and there is a
ticket,
create the ticket. ***Can't happen.
ENDIF

@Improvement Opportunity:

ARMS Process Report

- Use program status data structure to retrieve program name.

@Files (CRUD)

ZLIFM#1	(-R--)
CCEMP	(-R--)
CUSTMAST	(-R--)
DROFLF1	(-R--)
CCEXTCTL	(C---)
CCEXTDTL	(C---)
AMAPP	(C---)
CCINST	(C---)
CCSURDTL	(CR-D)
CCRES	(CRU-)
CCNUM	(-RU-)
CCMASTER	(CRU-)

@Notes:

- This programs only runs on the Central machine.
- The request(s) sent by the Rental Management Trading Partner can either be an authorization adds(AT-Add) or authorization changes(AT-Chg). There are three types of authorization adds: 1) An unsolicited authorization. 2) Authorization on a prior vendor authorization request. 3) Authorization for an open ticket (Vendor over-ride)
- Presently there is a business limitation where you can not authorize an existing open ticket.

Process

Hierarchical numeric ID: 1.1.1.1.5.2.2

Coded name: CC00EXV1

Name: PGM Interface CC with Rental Extension/Termination of Authorization (CC00EXV1)

Comment: @Purpose:

To update the claims connection database with any rental extension information received from the trading partner in the function group type 'EX'.

@Operational Method

- Wait indefinitely for the next DQAM61V1 data queue entry with a key of 'EXC'.

- Open programs files, if not previously opened.

- IF the receive data queue entry is a shutdown request, send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN

- Read the transaction data set records from AMAPP and populate the associated record format.
- Retrieve ARMS trading partner profile information through the use of ARMS data retrieval program(AM2090V1)
- IF extension is an extension request THEN

- Calculate new extension date.
- Generate associated text message for the history file.
- Re-calculate bill-to and bill-from dates.

ENDIF

- IF extension is a termination request THEN

ARMS Process Report

- Validate termination date received
- Generate associated text message for the history file.

ENDIF

ENDIF

@Files:

CCEXTCTL (CRU-)
CCEXTDTL (C---)
CCMASTER (-RU-)
AMAPP (-R--)
REQSTCTL (CRU-)

@Improvement Opportunity

- Use program status data structure to retrieve program name.

@Constants

CONST('Rental Extended Until:')
CONST('Rental Terminated On:')
CONST('OVRDBF FILE(CCMaster) WAITRCD(*IMMED)')
CONST('DLTOVR FILE(CCMaster)')
CONST('OVRDBF FILE(CCEXTCTL) WAITRCD(*IMMED)')
CONST('DLTOVR FILE(CCEXTCTL)')
CONST('OVRDBF FILE(CCEXTDTL) WAITRCD(*IMMED)')
CONST('DLTOVR FILE(CCEXTDTL)')

@Notes:

- This program only runs on the centralized financial system(Central).
- The request(s) sent by the Rental Management Trading Partner can either be an extension or termination.

Process

Hierarchical numeric ID: 1.1.1.1.5.2.3

Coded name: CC00CNV1

Name: PGM Interface CC with Rental Cancellation of Authorization (CC00CNV1)

Comment: @Purpose:

To process cancellation requests received from the trading partner in the functional group type 'CN'.

@Operational Method

- Wait indefinitely for the next DQAM61V1 data queue entry with a key of 'CNC'.

- Open programs files, if not previously opened.

- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN
 - Read the transaction data set records from AMAPP and populate the associated record format.
 - IF record not available THEN

ARMS Process Report

```
- End transaction request back to AM61V1.  
ELSE  
- Delete reservation record from file(CCRES).  
- Generate associated text message for the history file.  
ENDIF  
ENDIF
```

@Improvement Opportunity:

- Use program status data structure to retrieve program name

@Constants/Embedded Data

FILE() WAITRCD(*IMMED)

@Files:

AMAPP (-R--)
CCINST (CR--)
CCRES (-R-D)
CCDIRBIL (-R-D)

Process

Hierarchical numeric ID: 1.1.1.1.5.2.43
Coded name: VFYOBJ
Name: PGM Verify Objects Existence (VFYOBJ)
Comment: @Purpose:
To determine if the specified object exists on the host platform.
@Operational Method:
- Check for the objects existence, return a code of '1' when object not found.
@Notes:
- This program is used by programs written in languages other than the CL to take advantage of the CHKOBJ CL command.

Process

Hierarchical numeric ID: 1.1.1.1.6
Coded name:
Name: AUT Generate Vendor Authorization Maintenance Request (AC/RA/RC/RE/RN/TR/VM)
Comment: @Definition: The automatic process of dispatching data queue entries for appropriate programs to generate the required record formats(data set) to be sent to a Rental Management Trading Partner in response to an action by the Rental System or in response to some communication received from the Rental Management Trading Partner. (This includes ECARS and Claims Connection)

Process

Hierarchical numeric ID: 1.1.1.1.6.1
Coded name:
Name: AUT Generate Claims Connection Authorization Maintenance Request (AC/RA/RC/RE/RN/TR)

ARMS Process Report

Comment: @Definition: The automatic process of dispatching data queue entries for appropriate programs to generate the required record formats (data set) to be sent to a Rental Management Trading Partner in response to an action by the Rental System or in response to some communication received from the Rental Management Trading Partner.

Process

Hierarchical numeric ID: 1.1.1.1.6.1.1
Coded name: CC00TRV1
Name: PGM Generate CC Transfer Request (CC00TRV1)
Comment: @Purpose:

To generate authorized reservation transfer (functional group type AT-Transfer) group type transaction data sets for sending to another Rental application system in response to a Claims Connection Rental System user requesting a transfer of an authorized reservation to an Enterprise Rent-A-Car rental group's rental functional site/branch.

@Operational Method:

- Receive the next DQAM60V1 data queue entry with the key value of 'TRC' (for Claims Connection) in infinite wait mode.
- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.
- If the received data queue entry is a non-shutdown request, then from the data queue entry's Rental Location's Group ID and the Reservation ID passed, retrieve record from the CCRES rental reservation file.
- Populate the associated record formats' fields as follows:
 - Load the retrieved reservation's Transferred To Location Group/Branch ID to the COMD01 Comment Format's Comment Text with the COMTX data structure preformatted comment text immediately after the word "to".
 - Populate the ADJD01 data structure with the reservation's Bill-To Attention Name, Phone and Extension, the Bill-To Street Address, City, State and Postal Code.
 - If adjustor name or phone/extension fields of the ADJD01 record format are blank or zeroes, then retrieve the reservation's associated instructions file records for sequence 001 and extract and load this adjustor information from that instruction text line.
 - Load every non-blank reservation's associated CCINST Instruction file record's Instruction text field value to a COMD01 comment record format's comment field.
 - Load the RNTD01 Renter Detail and AUTD01 Authorization Detail record formats' fields with the associated fields' values of the retrieved CCRES file record.
 - If the retrieved CCRES file record's Repair Shop Name is not blank, then load that field value and its associated contact name and telephone number to the RPRD01 Repair Detail record format.
 - Load the retrieved CCRES file record's Renter's Car (Vehicle being replaced) Year (of Manufacture) and associated Make/Model and Date of Loss field values to their associated fields in the RNTD02 Renter Detail #2 record format.
 - Load the retrieved CCRES file record's Work Phone, Home Phone and Social Security Number field values to their associated fields in the RNTD02 Renter Detail #2 record format's fields.

ARMS Process Report

- If this transfer is NOT going to Claims Connection, transfer each associated electronic message detail file records field value to their associated CBKD01 record format's fields.

- If attempt to retrieve the associated CCEXTCTL Callback Control file record is successful and its record type is "R" (Reservation), transfer each of its Callback Type Codes and their associated Status Code field values to their associated CBKC01 Callback Control record format's fields. Likewise, if such a callback control format was loaded, then for every associated CCEXTDTL Callback Detail file record retrieved, transfer its Note #1 and Note #2 field values, along with the note's type original creation date, time, Employee ID and Program ID, to their associated CBKD01 Callback Detail record format's fields.

- Delete all associated CCSURDTL Surcharge Detail file records.
- Delete all associated CCEXTDTL Callback Detail file records.
- Delete the associated CCEXTCTL Callback Control file record.

- Write all of the non-blank record formats of the authorized reservation transfer (AT-Transfer) group type transaction data set to ANDIST transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.

- Sends the written authorized reservation transfer (AT-Transfer) group type transaction data set record formats' key value to the ANDIST output file's associated DQANDST data queue.

- Sends the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatch task has been completed.

@Files: (CRUD) [When is shown as (----), this is used as an externally-defined data structure.]

- ARMSDATA (----)
- ARMSKEY (----)
- DQAM60V1 (----)
- APPD01V1 (----)
- ADJD01V1 (----)
- AUTD01V1 (----)
- COMD01V1 (----)
- INSD01V1 (----)
- RNTD01V1 (----)
- RNTD02V1 (----)
- RPRD01V1 (----)
- CBKC01V1 (----)
- CBKD01V1 (----)
- \$#EMACH (----)
- CCINST (-R--)
- CCRES (-R--)
- CCSURDTL (-R-D)
- CCEXTCTL (-R-D)
- CCEXTDTL (-R-D)
- ANDIST (C---)

@Embedded Constant:

COMTX 'ARMS reservation transferred from Claims Connection to

@Notes:

ARMS Process Report

- This program does not use the INSD01 format, but the EC00TRV1 program does.

- This program is only executed as an ARMS never-ending batch program in the ARMS subsystem only on the distributed host computer system platform where Claims Connection Rental System application database resides. This program is currently submitted by the ARMS Start-Up Job (CLL810) when executed on the system known as "CENTRAL" host computer system platform. This never-ending batch program ends normally when a shutdown data queue entry is received as input.

Process

Hierarchical numeric ID: 1.1.1.1.6.1.2

Coded name: CC00RNV1

Name: PGM Generate CC Rental Notification (CC00RNV1)

Comment: @Purpose: To generate an opening or closing rental notification, functional group type 'RN' transaction data sets for sending to the Trading Partner host application system in response to a renter starting or ending the rental contract with a third-party rental vendor other than Enterprise Rent-A-Car.

@Operational Method:

- Receive the next DQAM60V1 data queue entry with the key value of 'RNC' in infinite wait mode.
- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.
- If the received data queue entry is a non-shutdown request with an action code of '3', then retrieve the record from the CCMaster open rental contract/ticket file and populate the associated record formats' fields, leaving the End Date and End Time fields initialized to zeroes.
- If the received data queue entry is a non-shutdown request with an action code of '4', then retrieve the record from the CCCLSC closed rental contract/ticket file and populate the associated record formats' fields, leaving the Start Date and Start Time fields initialized to zeroes.
- Load the OFFD01V1 format with the passed data queue entry's contract/ticket ID and its location, along with the retrieved shipping name, street address, city, state, postal code and toll-free telephone number for the Claims Connection office.
- Write Rental Notification (RN) group type transaction data set records to transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.
- Sends the written Rental Notification group type transaction set record formats' key value to the ANDIST output file's associated DQANDST data queue.
- Sends the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatch task has been completed.

@Files:

ARMS Process Report

- OFFDRB (-R--)
- CCMaster (-R--)
- CCCLSC (-R--)
- ANDIST (CRU-)

@Embedded Constant:

'7680' is the sole Group/Branch ID key value used to access its street/city/state address from the Branch Office Directory file (OFFDRB) at the beginning of the program.

8002277800 as the overriding telephone number loaded into all ELCD01:ELCPHN fields.

Process

Hierarchical numeric ID: 1.1.1.1.6.1.3

Coded name: CC00RCV1

Name: PGM Generate CC Request for Authorization Cancellation (CC00RCV1)

Comment: @Purpose:

To generate a requests for authorization cancellation, functional group type 'RC', transaction data sets for sending to the Trading Partner host application system.

@Operational Method:

- Receive the next DQAM60V1 data queue entry with the key value of 'RCC' in infinite wait mode.

- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.

- If the received data queue entry is a non-shutdown request with an action code of '6', then load the APPD01V1 record format's Add/Change/Transfer Code with "P", else load it with a blank. (Unsure this is currently needed since AM0046V1 does not reference any actions base on APPCODE with a "P" or blank value.)

- Write request for authorization cancellation (RC) group type transaction data set records to transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.

- Sends the written request for authorization cancellation (RC) group type transaction set record formats' key value to the ANDIST output file's associated DQANDST data queue.

- Sends the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatched task has been completed.

@Notes: This program is only executed as an ARMS never-ending batch program in the ARMS subsystem only on the host computer system platform where Claims Connection Rental System application database resides. This program is submitted by the ARMS Start-Up Job (CLL810) when executed on the "CENTRAL" host computer platform. This never-ending batch program ends normally when a shutdown data queue entry is received as input.

@Files:

- ANDIST (C---)

@Improvement Opportunity:

There is no real difference in this program and the CC00RCV1 program. With a minor change in the retrieved data queue entry key value's third position ("B" vs. "C") and the loading of the APPSRC field with that same value, there is no differences between these programs.

Process

Hierarchical numeric ID: 1.1.1.1.6.1.4

Coded name: CC00REV1

Name: PGM Generate CC Request for Authorization Extension (CC00REV1)

Comment: @Purpose:

To generate rental extension request, functional group type, 'RE' transaction data set for sending to the Trading Partner.

@Operational Method

```

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of
REC'.

- IF the receive data queue entry is a shutdown request, close program
files and send that request to data queue DQANDST to inform the distributed send
program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN
  - Retrieve the necessary data base records
  - IF extension request criteria met THEN
    - Build the record formats,
      write the record to the distribution transaction file
      (ANDIST).
    - Send a data queue entry on to the Distributed Send to
      Centralized data queue (DQANDST).
  ENDIF
- Send a data queue entry to Dispatch Data Queue
  (DQAM60V1) with a key of 'DIS' to finish the request.
ENDIF

```

@Files:

```

CCEXTCTL      (-R--)
CCEXTDTL      (-R--)
CCMASTER      (-R--)
REQSTCTL      (CRU-)
ANDIST        (C---)

```

@Notes:

- This program only runs on the centralized finicial system(Central).
- Application Data Set Formats
 - APPD01 - Enterprise internal format
 - VEDD01 - Vendor Extension format
 - COMD01 - Comment detail format
 - RPRD01 - Repair detail format

ARMS Process Report

Process

Hierarchical numeric ID: 1.1.1.1.6.1.5

Coded name: CC00RAV1

Name: PGM Generate CC Request for Authorization Maintenance (CC00RAV1)

Comment: @Purpose:

To generate an Authorization Request, functional group type, 'RA' transaction set for sending to the Trading Partner.

@Operational Method

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of 'RAC'.

- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN
 - Using the action code set the routing for the transaction
 - IF request is on a ticket THEN

- IF ticket is 'open' then

- Retrieve the open ticket data base records (CCMASTER)

- Retrieve the Rental Management Trading Partner profile specifics

- Build the applicable record format(s), writing the records to the distribution transaction

file(ANDIST).

(Only build RATD01 record format if trading partner profile indicates it).

ELSE

- Retrieve the closed ticket data base records (CCCLSC)

- Retrieve the Rental Management Trading Partner profile specifics

- Build the applicable record format(s), writing the records to the distribution transaction

file(ANDIST).

(Only build RATD01 record format if trading partner profile indicates it).

ENDIF

ELSE

- Retrieve the reservation data base records (CCRES)

- Retrieve the Rental Management Trading Partner profile specifics

- Build the applicable record format(s), writing the records to the distribution transaction file(ANDIST).

ENDIF

- Send a data queue entry on to the Distributed Send to Centralized data queue (DQANDST).

- Send a data queue entry to Dispatch Data Queue (DQAM60V1) with a key of 'DIS' to finish the request.

ENDIF

@Files:

ARMS Process Report

CCMASTER (-R--)
CCCLSC (-R--)
CCRES (-R--)
CCINST (-R--)
OFFDRB (-R--)
ANDIST (C--)

@Notes:

- This program only runs on the centralized financial system(Central).
- Application Data Set Formats
 - APPD01 - Enterprise internal format
 - AUTD01 - Authorization detail format
 - COMD01 - Comment detail format
 - RATD01 - Rate detail format
 - RNTD01 - Renter detail 1 format
 - RNTD02 - Renter detail 2 format
 - RPRD01 - Repair detail format
 - INSD01 - Insured detail format
 - ELCD01 - Arms vendor location

Process

Hierarchical numeric ID: 1.1.1.1.6.1.11

Coded name: CC00ACV1

Name: PGM Generate CC Authorization Confirmation (CC00ACV1)

Comment: @Purpose:

To generate an Authorization Confirmation, functional group type (AC), transaction set that informs the Trading Partner of the authorization confirmation number and the Enterprise location that is handling the insurance replacement.

@Operational Method:

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of 'ACC'.

- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN
 - Construct the necessary record format, write the record to the distribution transaction file(ANDIST)
 - Send a data queue entry on to the Distributed Send to Centralized data queue(DQANDST).
 - Send a data queue entry to Dispatch Data Queue(DQAM60V1) with a key of 'DIS' to finish the request.

@Improvement Opportunity:

- Use program status data structure to retrieve program name

@Embedded Constant:

'7680' - default location
'8002277800' - default location phone#
'CC00ACV1' - Program name

ARMS Process Report

@Files:

OFFDRB (-R--)
ANDIST (CRU-)

@Notes:

- This program only runs on the CENTRAL machine.
- Application Data Set Formats
 - APPD01 - Enterprise internal format
 - ELCD01 - Arms vendor location

Process

Hierarchical numeric ID: 1.1.1.1.6.2

Coded name:

Name: AUT Generate ECARS Authorization Maintenance Request (AC/RA/RC/RE/RN/TR/VM)

Comment: @Definition: The automatic process of dispatching data queue entries for appropriate programs to generate the required record formats(data set) to be sent to a Rental Management Trading Partner in response to an action by the Rental System or in response to some communication received from the Rental Management Trading Partner.

Process

Hierarchical numeric ID: 1.1.1.1.6.2.3

Coded name: EC00VMV1

Name: PGM Generate EC Vendor Message (EC00VMV1)

Comment: @Purpose:

To generate a vendor message functional group type, 'VM', transaction data sets for sending an electronic message from the rental agent to the trading partner.

@Operational Method:

- Receive the next DQAM60V1 data queue entry with the key value of 'VMB' in infinite wait mode.

- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.

- If the received data queue entry is a non-shutdown request, then load the APPD01V1 record format and for every RACMSD record read by the passed Group/Branch ID and the non-blank Ticket ID prefixed with a "D" (or if ticket is blank the passed Reservation ID prefixed with a "R"), load a COMD01V1 record format and update the RACMSD file record's MESSAGE STATUS field with the value 'O' (Output).

- Write vendor message (VM) group type transaction data set records to ANDIST transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.

- Send the written vendor message (VM) group type transaction set record formats' key value to the ANDIST output file's associated DQANDST data queue.

ARMS Process Report

- Send the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatched task has been completed.

@Files: (CRUD) (---- signifies usage as an external data structure)

- RACMSD1 (-RU-)
- ANDIST (C---)
- APPD01V1 (----)
- COMD01V1 (----)
- DQAM60V1 (----)

@Embedded Constant: Program name 'EC00VMV1'. (Could be replaced by use of the program status data structure's subfield for program.)

Process

Hierarchical numeric ID: 1.1.1.1.6.2.4

Coded name: EC00RCV1

Name: PGM Generate EC Request for Authorization Cancellation (EC00RCV1)

Comment: @Purpose:

To generate a request for authorization cancellation functional group type, 'RC', transaction data sets for sending to the trading partner.

@Operational Method:

- Receive the next DQAM60V1 data queue entry with the key value of 'RCB' in infinite wait mode.

- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.

- If the received data queue entry is a non-shutdown request with an action code of '6', then load the APPD01V1 record format's Add/Change/Transfer Code with "P", else load it with a blank. (Unsure this is currently needed since AM0046V1 does not reference any actions base on APPCODE with a "P" or blank value.)

- Write request for authorization cancellation (RC) group type transaction data set records to transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.

- Send the written request for authorization cancellation (RC) group type transaction set record formats' key value to the ANDIST output file's associated DQANDST data queue.

- Send the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatched task has been completed.

@Files: (CRUD)

- ANDIST (C---)

ARMS Process Report

@Improvement Opportunity: There is no real difference between this program and the program, CC00RCV1. With a minor change in the retrieved data queue entry key value's third position ("B" vs. "C") and the loading of the APPSRC field with that same value, there is no differences between these programs.

Process

Hierarchical numeric ID: 1.1.1.1.6.2.5
Coded name: EC00ACV1
Name: PGM Generate EC Authorization Confirmation (EC00ACV1)
Comment: @Purpose:

To generate an Authorization Confirmation, functional group type, 'AC', transaction data set that informs the Rental Management Trading Partner of the authorization confirmation number and the Enterprise location that is handling the insurance replacement.

@Operational Method

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of 'ACB'.

- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN
 - Construct the necessary record format, write the record to the distribution transaction file(ANDIST)
 - Send a data queue entry on to the Distributed Send to Centralized data queue(DQANDST).
 - Send a data queue entry to Dispatch Data Queue(DQAM60V1) with a key of 'DIS' to finish the request.

@Files (CRUD)
OFFDRB (-R--)
ANDIST (C---)

@Notes

- Application Data Set Formats
 - APPD01 - Enterprise internal format
 - ELCD01 - Arms vendor location

@Improvement Opportunity

- Use program status data structure to retrieve program name

@Constant/Embedded Data

'EC00ACV1' - Program name

Process

Hierarchical numeric ID: 1.1.1.1.6.2.6
Coded name: EC00REV1
Name: PGM Generate EC Authorization Extension Request (EC00REV1)
Comment: @Purpose:

To generate an Extension Request, functional group type 'RE', transaction data set for sending to the Rental Management Trading Partner in response to the

ARMS Process Report

Rental System speaking with the bodyshop and changing the contract's estimated completion date.

@Operational Method

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of 'REB'.

- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN

- Retrieve the necessary data base records

- IF extension request criteria met THEN

- Build the record formats,

update the database records, write the record to the distribution transaction file (ANDIST).

- Send a data queue entry on to the Distributed Send to Centralized data queue (DQANDST).

ENDIF

- Send a data queue entry to Dispatch Data Queue

(DQAM60V1) with a key of 'DIS' to finish the request.

ENDIF

@Notes

- Application Data Set Formats

- APPD01 - Enterprise internal format

- VEDD01 - Vendor Extension format

- COMD01 - Comment detail format

- RPRD01 - Repair detail format

@Files (CRUD)

CB007P00 (-R--)

CB012P01 (-RU-)

RACMSG2 (-RU-)

RACMAST (-R--)

RACSMAS (-R--)

ECEXTCTL (CRU-)

ANDIST (C---)

@Improvement Opportunity

- Use program status data structure to obtain the program name.

@Constants/Embedded Data

- 'EC00REV1' - pgm name

Process

Hierarchical numeric ID: 1.1.1.1.6.2.7

Coded name: EC00RAV1

Name: PGM Generate EC Authorization Maintenance Request (EC00RAV1)

Comment: @Purpose:

To generate an Authorization Request functional group type, 'RA' transaction data set for sending to the Rental Management Trading Partner in response to the Rental System creating/changing a rental contract.

ARMS Process Report

@Operational Method

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of 'RAB'.
- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.
- IF the receive data queue entry is a non-shutdown request THEN
 - Attempt to retrieve the reservation data base records (RACBRMST)
 - IF request is on a ticket THEN
 - Retrieve the open and closed ticket database records (RACMAST, RACSMASST, RACCLSC, RACSCLC)
 - Using the action code set the routing for the transaction
 - Retrieve the Rental Management Trading Partner profile specifics
 - Build the applicable record format(s) from the database files, writes the records to the distribution transaction file. (Only build RATD01 record format if profiled.)
 - Send a data queue entry on to the Distributed Send to Centralized data queue (DQANDST).
 - Send a data queue entry to Dispatch Data Queue (DQAM60V1) with key of 'DIS' to finish the request.
- ENDIF
- IF request is on a reservation THEN
 - Using the action code set the routing for the
 - Retrieve the Rental Management Trading Partner profile specifics
 - Build the applicable record format(s), writing the records to the distribution transaction file. (Only build RATD01 record format if profiled)
 - Send a data queue entry on to the Distributed Send to Centralized data queue (DQANDST).
 - Send a data queue entry to Dispatch Data Queue (DQAM60V1) with a key of 'DIS' to finish the request.

ENDIF

ENDIF

@Files	(CRUD)
RACMAST	(-R--)
RACSMASST	(-R--)
RACBRMST	(-R--)
RACBRIST	(-R--)
RACINS	(-R--)
RACBRWRT	(-R--)
OFFDRB	(-R--)
RACCLSC	(-R--)
RACSCLC	(-R--)
ANDIST	(C---)

Process

Hierarchical numeric ID: 1.1.1.1.6.2.8
Coded name: RTVEMCHA
Name: PGM Retrieve Machine Attributes (RTVEMCHA)
Comment: @Purpose:

To retrieve the machine attributes for the machine that this program is running on.

ARMS Process Report

@Operational Method:

- Retrieve the current system name (factor in emulation).
- Retrieve attributes associated to host platform.
- Return data elements to calling program.

@Notes:

- \$#EMCAH is an external data structure used as a template for the data passed to and from.

@Files (CRUD)
MACHID#1 (-R--)
GACTL (-R--)

Process

Hierarchical numeric ID: 1.1.1.1.6.2.17

Coded name: AM1070V1

Name: PGM Retrieve ARMS Vehicle Class and Rate Designation (AM1070V1)

Comment: @Purpose:

To retrieve the associate rental rate amount for a given vehicle class code, OR, to retrieve an associated vehicle class code for a specified rental rate amount.

@Operational Method:

The input parameters are: Group ID, Branch ID, Vehicle Class Code (Optional), Daily Rental Rate Amount (Optional), Effective Date (Optional - Else current system date will be used), and Control Mode ('T' = for rental contract ticket; 'R' = for rental Reservation).

- IF an Effective Date value is not received, then retrieve and use system date for Effective Date.

- IF request is on a rental contract/ticket (Control Mode = 'T'), a valid Vehicle Class Code will be returned for the effective date sent. (Daily Rental Rate Charged Amount will always have a value when sent since it is an open or closed rental contract.)

-- Validate the rate received by comparing effective date to the rate date ranges. Then retrieve the associated vehicle class code for that rate in the National Reservations Standard Rates (NRXRAT1) file, for "I" Insurance type customers only by Group/Branch IDs, Daily Rental Rate Amount and Effective Start Date, comparing the Effective Date to being on or within the rental rate amount's Effective Start and Stop Date ranges.

-- IF all of the records are read within a Group/Branch ID and no valid records are found, then retrieve a vehicle class code by reading backwards through NRXRAT1 until a valid record is found.

--- IF no valid record is found, then return blanks in the Vehicle Class Code.

--- IF a valid record is found, then verify its vehicle class code is valid by retrieving an associated record in National Reservations Car Type (NRCTY) file.

--- A valid record in the NRCTY file is a record where the Effective Date is within the Effective Start and Stop dates, inclusive.

--- IF rate and vehicle class do not match, then retrieve the class for the previous rate with a valid effective date.

- ENDIF

- IF request is on a reservation (Control Mode = 'R')

ARMS Process Report

-- When Vehicle Class Code and Daily Rental Rate Amount (Quoted/Authorized) has been received

--- Only validate that the Daily Rental Rate Amount (Quoted) received is compatible with the Vehicle Class Code by retrieving the daily rental rate amount's associated vehicle class code for that rate in the National Reservations Standard Rates (NRXRAT1) file, for "I" Insurance type customers only by Group/Branch IDs, Daily Rental Rate Amount and Effective Start Date, comparing the Effective Date to being on or within the rental rate amount's Effective Start and Stop Date ranges. Then find the associated vehicle class code for that daily rental rate amount.

--- IF the Daily Rental Rate Amount's Vehicle Class Code and the passed Vehicle Class Code do not match, then do not return either the rate nor vehicle class code, replace them with blanks.

- When Vehicle Class Code received, the associated Rate Amount will be returned if a valid record is found within the effective date.

- Validate the Vehicle Class Code received, then find the associated vehicle daily rental rate for that class.

IF rate and vehicle class do not match, then do not return either rate or class.

** This option is not return values when found. Issue will be written and submitted.

- When Rental Rate Amount received, an attempt will be made to find valid vehicle class.

the rate date ranges. Then find the associated vehicle class code for that rate.

IF rate and vehicle class do not match, then do not return either rate or class.

ENDIF

@Notes:

When indicating that Vehicle Class Code and Rental Rate Amount not returned, it means that these fields will be cleared to their lowest possible values, blanks and zeroes respectively.

Sometimes there are multiple vehicle types for any given Group ID, Branch ID, and Vehicle Class Code. However, this program has been changed to allow a rate to be returned correctly when this situation occurs.

@Files: (CRUD)

NRXRATES (-R--) by both logical access paths: NRXRAT1 and NRXRAT2.
NRCTYP (-R--) by both logical access paths: NRCTY and NRCTL2.

Process

Hierarchical numeric ID: 1.1.1.1.6.2.20

Coded name: ECCBBSM

Name: PGM Maintain Consolidated Body shop Callbacks Control File for ARMS (ECCBBSM)

Comment: @Purpose: To maintain the consolidated body shop callback information.

@Operational Method:

- This program accepts a parameter list that contains specific data elements as it relates to the callback (group, branch, ticket #, type, status, completion date, shop info, etc.) and updates the rental system (ECARS) data base file.

ARMS Process Report

passed:

- Determine tasks to be performed based on the control field value

calling program

- 'X' = Exit program
- 'C' = Retrieve callback information from database and return to
- 'D' = Delete callback
- 'K' = Check if callback exists
- 'L' = Lock callback from other users
- 'U' = Update callback with the passed information

@Files:

RACCBBS (CRUD)

Process

Hierarchical numeric ID: 1.1.1.1.6.2.22

Coded name: UTS001A

Name: PGM Generate Century for DOB (UTS001A)

Comment: @Purpose: To convert a six character date of birth to an eight character date with the correct century.

@Operational Method:

- this program accepts a parameter list from which a date is received as YYMMDD and converted to CCYYMMDD.
- Set base line date (current date minus 1)
- Load current date century into output date of birth century.
- IF the passed date of birth with the concatenated century is greater than the base line date, subtract 1 from the date of birth century.
- Return converted (CCYYMMDD) date of birth.

Process

Hierarchical numeric ID: 1.1.1.1.6.2.24

Coded name: EC00TRV1

Name: PGM Generate EC Transfer Request (EC00TRV1)

Comment: @Purpose:

To generate authorized reservation transfer (AT-Transfer) functional group type transaction data sets for sending to another Rental application system in response to a rental user transferring an authorized reservation to a different rental functional site (group).

@Operational Method:

- Receive the next DQAM60V1 data queue entry with the key value of 'TRB' (Transfers from Enterprise Branch Rental System) in infinite wait mode.
- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.
- If the received data queue entry is a non-shutdown request, then from the data queue entry's Rental Location's Group ID and the Reservation ID passed, retrieve record from the RACBRMST rental reservation file.

ARMS Process Report

- Populate the associated record formats' fields as follows:
 - IF the retrieved reservations closing Group/Branch ID is '7680' (Claims Connection), then load the COMD01 Comment Format's Comment Text with the COM002 data structure preformatted comment text and insert the reservation's Pickup Group/Branch immediately after the word "from", else load the COM003 data structure preformatted comment text with the inserted reservation's Pickup Group/Branch ID immediately after the word "from" and insert the reservation's closing Group/Branch ID after the word "to".
 - Populate the ADJD01 data structure with the reservation's Bill-To Attention Name, Phone and Extension, the Bill-To Street Address, City, State and Postal Code.
 - IF adjustor name or phone/extension fields of the ADJD01 record format are blank or zeroes, then retrieve the reservation's associated instructions file records for sequence 001 and extract and load this adjustor information from that instruction text line.
 - Load every non-blank reservation's associated instruction file record that has a sequence number 002 through 096 as a COMD01 comment record format.
 - Load the associated RACINS Renter Insurance Coverage file record's Policy ID and Policy Expiration date to the INSD01 Insured record format.
 - Load the retrieved RACBRIST file's 996 - 998 record's Insured name, address to the INSD01 record format fields.
 - IF the INSD01 record format's Policy ID and its expiration date is blank/zero, then load them from the RACBRIST file records' Policy ID and Expiration Date.
 - Load the RNTD01 Renter Detail and AUTD01 Authorization Detail record formats' fields with the associated fields' values of the retrieved RACBRMST file record.
 - If the retrieved RACBRMST file record's Repair Shop Name is not blank, then load that field value and its associated contact name and telephone number to the RPRD01 Repair Detail record format.
 - Attempt to retrieve the associated RACBRWRT Reservation Pre-Write file record and load its Date of Birth field value to the same field in the AUTD01 record format's same field.
 - If the retrieved RACBRMST file record's Renter's Car (Vehicle being replaced) Year (of Manufacture) is non-blank, then load that field value and its associated Make/Model and Date of Loss field values to their associated fields in the RNTD02 Renter Detail #2 record format.
 - If the retrieved RACBRMST file record's Work Phone, Home Phone or Social Security Number field values are not zeros or blanks, then also load those values to their associated fields in the RNTD02 Renter Detail #2 record format.
 - For every associated RACBDET Callback Detail file record retrieved, transfer its Note #1 and Note #2 field values, along with the note's type original creation date, time, Employee ID and Program ID, to their associated CBKD01 Callback Detail record format's fields.
 - If this transfer is NOT going to Claims Connection, transfer each associated electronic message detail file records field value to their associated CBKD01 record format's fields.
 - If attempt to retrieve the associated CB007P00 Callback Control file record is successful, transfer each of its Callback Type Codes and their associated Status Code field values to their associated CBKC01 Callback Control record format's fields.
 - Write all of the non-blank record formats of the authorized reservation transfer (AT-Transfer) group type transaction data set to ANDIST transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.
 - Sends the written authorized reservation transfer (AT-Transfer) group type transaction data set record formats' key value to the ANDIST output file's associated DQANDST data queue.

ARMS Process Report

- Delete all associated ECSURCHG Surcharge Detail file records.
- Delete all associated RACMSC Electronic Message Control file records.
- Delete the associated CB012P Electronic Message Detail file record.
- Delete the associated RACCBK3 Callback Consolidation Skeleton file

record.

- Delete all associated RACCBDET Callback Detail file records.
- Delete all associated RACPAYOP Open Payments file records.

- Sends the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatch task has been completed.

@Notes:

Do not write INSD01 format for a transfer of an authorized reservation for a claimant, since the AT-Add had updated the claimant's insurance information to create a RACINS (ECR Renter's Insurance Detail) record without loading the INSURED NAME and CITY/ST/ZIP. However, when the branch reservation is transferred, this program loaded the INSD01 record format with this insured information from RACINS but the INSURED NAME, CITY/ST/ZIP and RECORD FORMAT ID were not loaded since they are blanks. This is because that information is not available at the time of the transfer. This is to prevent the AT-Transfer from being rejected in AM0025V1.

@Files: (CRUD) [When is shown as (---), this is used as an externally-defined data structure.]

- ARMSDATA	(---)
- ARMSKEY	(---)
- DQAM60V1	(---)
- APPD01V1	(---)
- ADJD01V1	(---)
- AUTD01V1	(---)
- COMD01V1	(---)
- INSD01V1	(---)
- RNTD01V1	(---)
- RNTD02V1	(---)
- RPRD01V1	(---)
- CBKC01V1	(---)
- CBKD01V1	(---)
- \$EMACH	(---)
- RACBRMST	(-R-) Opened with the OVRDBF
- RACBRIST	(-R-) Opened with the OVRDBF
- RACINS	(-R-) Opened with the OVRDBF
- RACBRWRT	(-R-) Opened with the OVRDBF
- RACCBK3	(-R-D) Opened with the OVRDBF
- RACBCOC	(-R-D) Opened with the OVRDBF
- RACPAYOP	(-R-D) Opened with the OVRDBF
- ECSURCHG	(-R-D) Opened with the OVRDBF
- CB007P00	(-R-D) Opened with the OVRDBF
- RACCBDET	(-R-D) Opened with the OVRDBF
- RACMSC	(-R-D) Opened with the OVRDBF
- CB012P00	(-R-D) Opened with the OVRDBF
- RACPAYCL	(C---) Opened with the OVRDBF
- ANDIST	(C---) Opened with the OVRDBF

@Constants as Compile-Time Data:

OVRDBF FILE() WAITRCD(*IMMED)

ARMS Process Report

YES	C	CONST('Y')
NO	C	CONST('N')
OFF	C	CONST('0')
ON	C	CONST('1')
Process/shutdown codes		
TR	C	CONST('TR')
SD	C	CONST('SD')
TSTDWN	C	CONST('*DOWN')
Format Names		
APPDF1	C	CONST('APPD01')
ADJDF1	C	CONST('ADJD01')
AUTDF1	C	CONST('AUTD01')
INSDF1	C	CONST('INSD01')
RNTDF1	C	CONST('RNTD01')
RNTDF2	C	CONST('RNTD02')
RPRDF1	C	CONST('RPRD01')
COMDF1	C	CONST('COMD01')

Pre-formatted comment text:

COM002	'ARMS reservation transferred from	to Claims
COM003	'ARMS reservation transferred from	to

Process

Hierarchical numeric ID: 1.1.1.1.6.2.25
 Coded name: EC00RNV1
 Name: PGM Generate EC Rental Notification (EC00RNV1)
 Comment: @Purpose:

To generate an opening or a closing rental notification functional group type, 'RN', transaction data sets for sending to a Rental Management Trading Partner host application system.

@Operational Method:

- Receive the next DQAM60V1 data queue entry with the key value of 'RCB' in infinite wait mode.
- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.
- If the received data queue entry is a non-shutdown request with an action code of '3', then attempt to retrieve the record from the RACMAST open rental contract/ticket file first. If not found attempt to retrieve from QRACCLSQ. Populate the associated record formats' fields including loading the Contract Date to Start Date field and Time Out of Rental Segment #1 to the Start Time field, leaving the End Date and End Time fields initialized to zeroes to signify this is an opening rental notification.
- If the received data queue entry is a non-shutdown request with an action code of '4', then retrieve the record from the QRACCLSQ closed rental contract/ticket file and populate the associated record formats' fields including the End Date and End Time fields, leaving the Start Date and Start Time fields initialized to zeroes to signify this is a closing rental notification.
- If the received data queue entry is a non-shutdown request with an action code of '5', then attempt to retrieve the record from the RACMAST open rental contract/ticket file first. If not found attempt to retrieve from QRACCLSQ.

ARMS Process Report

Populate the associated record formats' fields including End Date and End Time fields, leaving the Start Date and Start Time fields initialized to zeroes to signify this is a closing rental notification.

- If the received data queue entry is a non-shutdown request with an action code of other than the previously specified values or there was no successful database retrieval, then generate a formatted program dump, call/execute AM0097 program to notify the ARMS On-Call of this failure and reject the transaction for further processing.

- Load the OFFD01V1 format with the passed data queue entry's contract/ticket ID and its location, along with the retrieved shipping name, street address, city, state, postal code and the telephone number for that Enterprise Rent-A-Car rental office.

- Write Rental Notification (RN) group type transaction data set records to transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.

- Send the written Rental Notification group type transaction set record formats' key value to the ANDIST output file's associated DQANDST data queue.

- Sends the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatch task has been completed.

@Files: (CRUD)

- OFFDRB (-R--)
- RACMAST (-R--)
- QRACCLSQ (-R--)
- ANDIST (C---)

@Constants:

'7680' is the sole Group/Branch ID key value used to access its street/city/state address from the Branch Office Directory file (OFFDRB) at the beginning of the program.

8002277800 as the overriding telephone number loaded into all ELCD01:ELCPHN fields.

Process

Hierarchical numeric ID: 1.1.1.1.7

Coded name:

Name: AUT Send Transaction for Centralized Processing (AM106 - AM105)

Comment: @Definition: The automatic process of sending the transaction data set from the distributed machine to the centralized machine for processing, via ICF files.

Process

Hierarchical numeric ID: 1.1.1.1.7.1

Coded name: AM0106

Name: PGM Send Transaction for Centralized Processing (AM0106)

Comment: @Purpose:

To send transaction data sets from the Distributed machine to the Centralized machine for processing.

@Operational Method:

- Acquire the program device 'AM0106'
- Wait indefinitely for the next DQANDST data queue entry
- IF the receive data queue entry is a shutdown request
 - Increment counter. When the count equals the number of ARMS never ending transaction programs running on that host platform close program files and send that request to ICF device AM0106 to inform the centralized receive program that AM0106 has ended.
 - Submit job for error report and history file update(CLL797).
- ENDIF
- IF the receive data queue entry is a non-shutdown request
 - Retrieve the application transaction data set records
 - Write the data set to ICF device AM0106 to send transaction data set to the centralized ARMS host platform
- ENDIF.

@Notes: This program is currently evoked from the executing Receive Transaction from Distributed Host System (AM0105) program on the centralized ARMS host platform to be the source of the sender linked ICF communication session. Presently, the distributed rental system host platform are all systems excluding BIRELAND, VGERMANY, and UK.

@Files: (CRUD)

- ANDIST (-R--)
- AMVRGTST (-R--)

Process

Hierarchical numeric ID: 1.1.1.1.7.2

Coded name: AM0105

Name: PGM Receive Transaction from Distributed Host System (AM0105)

Comment: @Purpose:

To receive transaction data sets from the distributed rental system host platform on the ARMS centralized host platform.

@Operational Method:

- When starting the program, attempt to acquire the device up to 5 times with a one minute delay between each try before signalling an error.
- Receive input transaction from the distributed rental system host platform
- IF data was received
 - IF transaction data set is equal to Vendor Error(ER) or Vendor Office Information(OF)
 - Write records to pre-package file(AMPACK)
 - Send data queue entry to pre-package (DQAMPKG)
 - ELSE
 - Write records to Edit Transaction file(AMSET)
 - Send data queue entry to Transaction Editor (DQAM25)
- ENDIF

ARMS Process Report

ELSE

- IF shutdown request receive
 - Send data queue entry to Transaction Editor(DQAM25)

ENDIF

- Release Device

ENDIF

@Files: (CRUD)

- AMPACK (C---)
- AMSET (C---)

@Notes: It is currently submitted by the Start-Up ARMS Jobs (CLL810) program on the centralized ARMS host platform to be executed with a single 1-character input parameter of the distributed System (Machine) ID to start linked ICF communication session. Presently, the distributed rental system host platform are all systems excluding BIRELAND, VGERMANY, and UK.

Process

Hierarchical numeric ID: 1.1.1.1.7.3

Coded name:

Name: DTQ Input to AM106 (DQANDST)

Comment: @Definition: DQANDST is a data queue used to provide input to PGM AM0106 which sends transactions from the distributed machine to the centralized processing machine.

Process

Hierarchical numeric ID: 1.1.1.1.7.7

Coded name: CLL797

Name: PGM Completed Errors Report (CLL797)

Comment: @Purpose: This program is one of the programs along with AM0797 and AM0099 that were intended to implement Tom Stratton's error handler system for error capture and response.

@Operational Method:

- Read files ANDQER and AMERDT to gather the details for any error that was marked as completed in AM0099.

@Notes: These programs are not used at this time even though they are on production systems. Essentially this was an implementation failure due to complexity and user friendliness issues.

Process

Hierarchical numeric ID: 1.1.1.1.8

Coded name:

Name: AUT Package Transaction into Transmission (AM120)

Comment: @Definition: The automatic process of preparing a transaction for transmission by putting it in a transmission envelope.

Process

Hierarchical numeric ID: 1.1.1.1.8.1

ARMS Process Report

Coded name: AM0120

Name: PGM Package Transaction (AM0120)

Comment: @Purpose: To add control start and end record formats to the transaction data sets to create an enveloped transmission. These control formats are used to route the transmission and indicate the type of transaction set being sent.

@Operational Method:

- Receive the next communication acknowledgement, authorization maintenance request, rental maintenance recognition or authorization maintenance error from data queue DQAMPKG until a shutdown request is received.
- Upon receipt of shutdown requests, pass request to time line DQAM70V1.
- For each transaction data set
 - Build the control start record formats(Transmission, Group & Set) and write record formats to send file(AMSEND)
 - For each transaction data set record format
 - Translate the location(AUTD01/ELCD01/IEBH01)
 - Write record format to send file(AMSEND)
 - Build the control end record formats(Set, Group, & Transmission) and write record formats to send file(AMSEND)
 - Send a data queue to the connect-specific communications sender

@Files: (CRUD)

- AMPACK (-R--)
- ARMSPR3 (-R--)
- AMXBCO (-R--)
- AMSEND (C---)

Process

Hierarchical numeric ID: 1.1.1.1.8.6

Coded name: DQAMPKG

Name: DTQ Input for Program AM0120 (DQAMPKG)

Comment: @Definition: DQAMPKG is a data queue used to provide input to PGM AM0120 which packages a transaction in preparation to sending it out to an EDI Trading Partner.

Process

Hierarchical numeric ID: 1.1.1.1.9

Coded name:

Name: AUT Dispatch Vendor Authorization Maintenance Request

Comment: @Definition: The automatic process of updating the ARMS database on the distributed machine and dispatching the data queue entry for the appropriate format generator program in response to some action by the Rental System.

Process

Hierarchical numeric ID: 1.1.1.1.9.1

Coded name: AM0060V1

Name: PGM Dispatch Rental Systems Request (AM0060V1)

Comment: @Purpose:

ARMS Process Report

To accept requests from the rental systems to maintain the distributed ARMS Rental Transaction Cross-Reference and dispatch transaction set generation requests.

@Operational Method:

- Override the printer file outq to ARMSDUMP so that the dump can be retained for as long as needed. The default outq ERRDUMPS is cleared daily.

- Receive authorization management or rental notification requests (AC, IN, RA, RC, RE, Opening RN, Closing RN, TR and VM) from the RENTAL SYSTEMS. All requests or notifications as output from the rental systems with a key of 'DIS' will be read from the input data queue (DQAM60V1) by this program.

- Based on externally described rules (AMPGMTBL, AMPGMRUL and AMPGMCTL), dispatch the request to the appropriate request handler process (format generator program e.g., EC00RAV1) dispatching an output data queue entry that has been modified from the input data queue entry to the correct rental system transaction set generator never-ending program/job, based on the key information found in the AMTRNCTL file (ACB, ACC, RAB, RAC, RCB, RCC, REB, REC, RNB, RNC).

- A transaction set generation request may lead to another request. After each request is sent out, a feedback entry to this program informs that the dispatched request has been completed.

- Edit to check for valid Program ID, Rule ID and dispatch execution condition. Generate an error message for an invalid dispatch execution condition. No page is generated for invalid dispatch execution condition. However, for invalid Program ID or Rule ID, ARMS On-Call pager will be sent an informational message. In all cases, transaction request is rejected.

- Receive and process environment change requests which allow the program to dynamically adapt for load balancing during heavy transmission load (Group Type = 'WT') or to shutdown the format generators and itself.

- Log the received data queue entry for audit and problem resolution.

- Retrieve the information required to process the request via appropriate retrieval module (EC2090V1 for ECARS and CC2090V1 for Claims Connection requests). If the retrieval program detects a lock on the ticket, defer processing the request.

- A request may need to be processed in more than one pass through the program. The current step being processed is logged externally (in AMTRNCTL). All requests related to this transaction id are deferred while a step for the transaction id is in progress. When the logged request has been completely processed, the log for that transaction is removed so that the deferred requests can now be processed.

@Files:

ARMSPR1	(-R--)
AMPGMTBL	(-R--)
AMPGMRUL	(-R--)
AMPGMCTL	(-R--)
AMTRNCTL	(CR-D)
AMXBCO	(-R--)
AMVRGTST	(-R--)
AMQUETBL	(-R--)
AMSTSX	(-R--)
AM0060LG	(C---)

ARMS Process Report

ARMSKEY (-R--)
ANDSG1V1 (-R--)
\$#EMACH (-R--)
AMPGMCTL (-R--)
DQAM60V1 (-R--)
DQAM60V1 (-R--)

@Notes:

- AC (Authorization Confirmation)
- IN (Invoice)
- RA Add (Request to Authorize a reservation or an open ticket)
- RA Change (Request to Change an Authorization for a reservation or an open ticket)
- RC (Request to Cancel an authorization)
- RE (Request to Extend a rental)
- Opening RN (Rental Notification of an OPEN ticket)
- Closing RN (Rental Notification of a CLOSED ticket)
- TR (Request to Transfer Authorization from one location to another)

Process

Hierarchical numeric ID: 1.1.1.1.9.10
Coded name: DQAM6AV1
Name: DTQ Secondary Input to AM0060 (DQAM6AV1)
Comment: @Definition: DQAM6AV1 is a data queue used as a backup queue for PGM AM0060V1 when it receives a request that can not be processed due to work in process as represented by a record in file AMTRNCTL. This secondary queue is periodically checked when there has been a set period of inactivity in the primary data queue for PGM AM0060V1.

Process

Hierarchical numeric ID: 1.1.1.1.9.12
Coded name: DQAM60V1
Name: DTQ Primary Input for AM0060 (DQAM60V1)
Comment: @Definition: DQAM60V1 is a data queue used to provide the primary source of input to PGM AM0060V1 which handles processing needed as a result of requests and or actions taken by the Rental System (ECARS) and routes requests for formats to be generated to the correct format generator programs.

Process

Hierarchical numeric ID: 1.1.1.1.9.13
Coded name: EC2090V1
Name: PGM Retrieve ECARS Data (EC2090V1)
Comment: @Purpose:

To retrieve specified information from the ECARS rental database.

@Operational Method:

- If the passed ARMS DS Identification code is 'G1', attempt to retrieve the data otherwise if the code is 'GO', return to calling program.
- If the reservation number is passed, get the reservation data.

ARMS Process Report

- If the ticket number is passed, get the ticket data.

@Files:

- RACBRMAST	(-R--)
- NR025L01	(-R--)
- RACMAST	(-R--)
- RACSMAS	(-R--)
- QRACCLSQ	(-R--)
- RACSCLD	(-R--)
- RACPENDC	(-R--)
- ANDSG1V1	(-R--)
- DTPARM	(-R--)

@Notes:

A reservation created by National Reservations has the routed Machine ID as the first character of the RESERVATION ID.

This program is called by a variety of non-ECARS programs. Each program calling this program may have a different parameter list associated with the program call. Each different parameter list, defines a different type of function required by this program.

Functions desired for this program must be closely coordinated with any NON-ECARS rental application development staff. All objects involved with any desired function must be moved into production at the same time.

This program has been designed and written to be as modular as possible. All logic for each type of transaction is contained in a single subroutine. The required files are opened at the beginning of each subroutine, the other files are not opened.

@Improvement Opportunities:

- 1.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.
- 2.) Delete the constant 'EC2090V1' and replace its usage with the Program Status Data Structure data element for PROGRAM ID.
- 3.) Convert program from OPM RPG to ILE RPG.

Process

Hierarchical numeric ID: 1.1.1.1.9.14
Coded name: CC2090V1
Name: PGM Retrieve Claims Connection Data (CC2090V1)
Comment: @Purpose:

To retrieve specified rental information from the Claims Connection database.

@Operational Method:

- Retrieve the Claims Connection reservation or ticket data from the CC database and return the information using pre-defined external data structure.

ARMS Process Report

@Files:

CCMASTER (-R--)
CCCLSC (-R--)
CCRES (-R--)
ANDSG1V1 (-R--) EDS

@Note: When a ticket is opened, the reservation is deleted from the reservation master file.

@Improvement Opportunities:

- 1.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.
- 2.) Delete the constant 'CC2090V1' and replace its usage with the Program Status Data Structure data element for PROGRAM ID.
- 3.) Convert program from OPM RPG to ILE RPG.

Process

Hierarchical numeric ID: 1.1.1.1.9.15
Coded name: CLDQINFO
Name: PGM Retrieve Data Queue Information (CLDQINFO)
Comment: @Purpose:

This program receives a parameter and using the Application Program Interface (API) QMHQRDQ, retrieves a data queue description.

@Operational Method:

- Prepare the parameters used by the API then call the API
- Parse each attribute retrieved:
 - Length (binary)
 - Key length (binary)
 - Sequence type (FIFO, LIFO, KEYED)
 - Sender ID (Y or N)
 - Force Indicators (Y or N)
 - Description
 - Number of messages on the queue (binary)
 - Max. number of message on queue (binary - not used)
- Convert binary values to decimal
- Return to values to calling program.

Process

Hierarchical numeric ID: 1.1.1.1.18
Coded name:
Name: AUT Dispatch Customer Authorization Maintenance Request
Comment: @Definition: The automatic process of updating the ARMS database on the distributed machine and dispatching the data queue entry for the appropriate program to then update the Rental database.

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.1.1.18.3
Coded name: AM0061V1
Name: PGM Edit and Dispatch Customer Request (AM0061V1)
Comment: @Purpose:

To perform the distributed Edits and Dispatch a Customer Request transaction to the appropriate Rental System never-ending program/job or Financial System Interface daily batch transaction files.

@Operational Method:

- Execute as a never ending program on the distributed host machine.
- Receive the next data queue entry with a key value of 'DIS' from data queue DQAM61V1 or DQAM6BV1.
- If a 'WT' (Wait Time) group type is received, change the amount of time the primary data queue must elapse before switching to the secondary data queue to the value in group type. However if the value = 99999, set the wait time to -1, infinite.
- If a 'DF' (Deferred) group type is received, execute the program to return the number of entries in secondary data queue.
- If a 'SD' (Shutdown Request) group type is received, 1 pass is made through the secondary data queue. Any entry not processed will remain deferred. After passing through the secondary data queue, generate shutdown request for the necessary rental interface program(s).
- If an 'RM' (Remittance Advice) group type is received at the distributed host machine that is not the financial systems host, only update the cross-reference status.
- If a daily rental transaction (excluding RM) is received at the distributed host machine that is the financial systems host, the program dumps and invokes error logging process, AM0098.
- If the authorizing customer is different from the original customer on an old style vendor override, generate a request for cancellation for the original customer who received the original request for authorization.
- After business rules have been validated, the request is either rejected back to the trading partner, passed on the appropriate interface request handler process, and in certain cases the request is deferred. (A key comprised of the transmission group type and business type is created when passing the request to the interface program)
- Log the rejection of a request for authorization for an open ticket (in file AMSTSX).
- If the ticket referenced in the vendor override request is not found, generate standardized comment stating that the vendor override failed (data area AMATTCDA, format COMD01 in file AMAPP).

@Files:

AMAPP	(CRU-)
AMVRGTL1	(-R--)
AMQUETBL	(-R--)
AMSTSX	(CR--)

@Notes:

- AT (Authorization Add/Change/Transfer)
- CC (Customer Confirmation)
- CN (Customer Cancellation Request)
- CM (Customer Message)
- EX (Rental Extension)
- PM (Customer Payment Advice)
- RM (Customer Remittance Advice)

The printer file for error DUMPS QPPGMDMP is overridden in this program to go to outq ARSMDUMP instead of the Enterprise default of ERRDUMPS because the outq ERRDUMPS is cleared on a daily basis and sometimes the ARMS on call staff can not get around to fix the problem under consideration for more than a day. This way the dump is retained until explicitly deleted.

***** External Data Structures *****

ANDSG1V1
 ARMSKEY
 APPD01V1
 CRED01V1
 COMD01V1
 PMTD01V1
 RMTD01V1
 DQAM60V1
 AMATTCDA (-R--) Data Area
 \$#EMACH

Process

Hierarchical numeric ID: 1.1.1.1.18.4

Coded name:

Name: DTQ Secondary (Deferred) Input to AM0061 (DQAM6BV1)

Comment: @Definition: DQAM6BV1 is a data queue that is used as a source of deferred input to PGM AM0061V1 which routes Trading Partner transactions to the correct Rental System interface program. This data queue is input to by EC00ATV1 and AM0061V1.

Process

Hierarchical numeric ID: 1.1.1.1.18.5

Coded name: DQAM61V1

Name: DTQ Primary Input to AM0061 and Rental System Interfaces (DQAM61V1)

Comment: @Definition: DQAM61V1 is a data queue used to provide input to PGM AM0061V1 and Rental System Interface programs which process inbound transaction requests from the EDI Trading Partner.

Process

Hierarchical numeric ID: 1.1.1.1.18.8

Coded name: AM0049V1

Name: PGM Update Financial Services Files with Remittance Detail (AM0049V1)

Comment: @Purpose:

To write or update records to batch transaction header and detail holding files that are used to update the FINANCIAL SYSTEMS' Cash Receipts application database files with the detailed information of EFT remittance advice detail transaction data sets that were sent and processed through the ARMS system. This

ARMS Process Report

information will be used by Financial Systems to process Electronic Funds Transfer and to distribute and apply the incoming funds. There will be one header record for each EFT batch received.

@Operational Method:

Using the passed input parameter field values for COMPANY PROFILE ID for Ins.ID, EFT IDENTIFICATION, and a SEQUENCE NBR. starting at 1 through 9 as a key, attempt to access for update an existing associated Cash Receipts EFT Remittance Advice Header File (CREFTH) record.

IF no record is found, write a new record, using this key value. Load the passed input parameter PAYMENT AMOUNT field value to the output BATCH TOTAL field and load GROUP ID with "76" and BRANCH ID with "99" and load the RECORD ADDED audit fields appropriately.

IF a record is found, check its STATUS, being one of the following values:

blank	=	Unprocessed
I	=	In use by another user
P	=	Pending
S	=	Submitted to batch
O	=	Processed or Posted
D	=	Deleted

IF this batch record's STATUS is submitted ("S"), posted ("O"), or, deleted ("D"), then the record is unavailable to ARMS interface. Increment the key's SEQUENCE NBR. by 1 and reattempt update access. However, IF there were 9 records found that all were unavailable, then DUMP and notify ARMS On-Call of the exception error by calling 'AMPSSR' with the passed input parameter field being a concatenation of the passed input parameter fields' values to this program.

ELSE, update this record's accumulated BATCH TOTAL with the passed input parameter field, PAYMENT AMOUNT.

IF no exception error has occurred, then attempt to access for update an existing associated Cash Receipts EFT Remittance Advice Detail File (CREFTD) record, using the same key values last used for the Header File.

IF no record is found, write a new record, using this key value and a LINE NBR value of 1. Load the passed input parameter field, PAYMENT AMOUNT value to the output PAYMENT TOTAL field and load the passed GROUP ID, BRANCH ID and TICKET ID prefixed with "D" and load the RECORD ADDED audit fields appropriately.

IF a record is retrieved, compare the retrieved record's GROUP ID, BRANCH ID, and TICKET ID to the passed GROUP ID, BRANCH ID and TICKET ID prefixed with "D".

IF they match, then add the passed input parameter PAYMENT AMOUNT field value to the current PAYMENT AMOUNT field and update the RECORD CHANGED audit fields appropriately.

IF no records are found whose GROUP ID, BRANCH ID and TICKET ID match, then write a record to this file using the same key fields' values and set the LINE NBR. to 1 or 1 more than was last retrieved. Load the passed passed GROUP ID, BRANCH ID and TICKET ID prefixed with "D", ENTERPRISE INTERNAL CUSTOMER ID, PAYMENT AMOUNT to their associated fields and load the RECORD ADDED audit fields appropriately.

@Files: (CRUD)

ARMS Process Report

CREFTH (CRU-) Cash Receipts EFT Payment Remittance Advice Header File,
by , by Ins. ID, EFT ID#, and Sequence Nbr.

CREFTD (CRU-) Cash Receipts EFT Payment Remittance Advice Detail File,
by Ins. ID, EFT ID#, Sequence Nbr., and Line Number.

@Notes:

This program is called by the ARMS Distributed Edits and Dispatch
(AM0061V1) program with the following input parameter fields:

5	Alphameric	Company Profile ID
20	Alphameric	EFT Identification
2	Alphameric	Group ID
2	Alphameric	Branch ID
6	Alphameric	Rental Contract/TICKET ID
7	Alphameric	Enterprise Internal Customer ID
15/5	Numeric	Payment Amount

For any program exception error, the *PSSR - ERROR HANDLING SUBROUTINE
will be executed that will output a spooled RPG formatted DUMP and CALL 'AMPSSR' with
the passed input parameter field being a concatenation of the passed input parameter
fields' values to this program to notify the ARMS On-Call staff.

The ARMS system will now enable electronic invoices, payments and
remittances (processed here) to be sent/received for non-ARMS Rental Management
Trading Partners that are Direct Billing/Remittance Trading Partners, such as
Tennessee Valley Authority (an Enterprise Fleet Services customer). Therefore, ARMS
will receive a data queue entry to generate an electronic invoice (and create an
associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) when the
trading partner never sent to ERAC (via ARMS) any initial rental authorization.

@Embedded Data/Constants:

'AM0049V1' is used as a literal constant to pass as an input parameter
field value to the calls of 'AMPSSR' program and the output of the PROGRAM ID audit
field in the CREFTD and CREFTH files' records.

'7699' (Host) is always used to load the GROUP and BRANCH ID fields of
the Cash Receipts EFT Payment Remittance Advice Header File

@Improvement Opportunities:

Replace the literal constant 'AM0049V1' with the Program Status Data
Structure's data element for PROGRAM ID in the RECORD ADDED and RECORD CHANGED audit
fields.

Process

Hierarchical numeric ID: 1.1.1.1.19

Coded name:

Name: AUT Send Transmission to Trading Partner Company

Comment: @Definition: The automatic process of translating (if necessary) and
sending a transmission to the Trading Partner through one of several EDI mediums.

Process

Hierarchical numeric ID: 1.1.1.1.19.2

Coded name: EDZSCWA

ARMS Process Report

Name: PGM Check for Wrapped Transactions to Send (EDZSCWA)

Comment: @Purpose:

To check for Outbound wrapped EDI transactions for Any EDI integration software sender processing. These transactions have been wrapped by program EDLSIMA in collaboration with program AMZSIMA.

@Operational Method:

This program receives a data queue entry keyed by the trading partner communications port id and if it is not a shutdown entry, it will read the Any EDI integration software Connection Log file to determine if there are any outbound wrapped messages ready to be sent for this trading partner.

Based on the above information the return parameters are set.

@Files:

EXLLCPP (in library EXTSYSF) (-R--)

Process

Hierarchical numeric ID: 1.1.1.1.19.7

Coded name: AMSCPIC

Name: PGM Send Proprietary Transmission to an ARMS-connected VAN (AMSCPIC)

Comment: @Purpose:

To request a program start communication conversation from an ARMS trading partner, then continuously sends proprietary EDI data set transmissions to the trading partner. These transmissions are triggered by the receipt of their key value via an input data queue entry until a data queue entry of *DOWN is received from the input data queue.

@Operational Method:

For any error exception condition that occurs (including CPI-C errors), output a spooled RPG formatted dump and CALL 'AMPSSR to notify ARMS On-Call staff of the condition and end the program.

At program startup:

- Derive ARMS to an ARMS-connected VAN Transmissions to be sent input data queue name by concatenating 'DQ' with input parameter COMPANY PROFILE ID value (example 'DQVN101').

- Call QCMDEXC to execute the ADDLIBLE QSYS2 command to access the OS/400 System CPI-C objects.

After startup has completed, repetitively perform the following steps:

- Receive the next data queue entry from the derived input data queue into the ARMSKEY external data structure, waiting infinitely for an entry.

- IF after any of the subsequent program executions, the returned PROGRAM RETURN CODE is not equal to 0 (zero), then execute the program exception error routine.

- Initialize a CPI-C communication conversation by executing 'CMINIT' program with the following parameters:

ARMS Process Report

8 Character (Output) CONVERSATION ID
5 Character (Input) CONVERSATION SYMBOLIC DESTINATION ID
(CSI), loaded with the input parameter COMPANY PROFILE ID.
9/0 Binary (Output) PROGRAM RETURN CODE

- Allocate a CPI-C communication conversation by executing 'CMALLC' program with the following parameters:

8 Character (Input) CONVERSATION ID
9/0 Binary (Output) PROGRAM RETURN CODE

- Set a CPI-C communication session/conversation send type by executing 'CMSST' program with the following parameters:

8 Character (Input) CONVERSATION ID
9/0 Binary (Input) CONVERSATION SEND SET TYPE, loaded with the value 1.
9/0 Binary (Output) PROGRAM RETURN CODE

- IF the received data queue entry's COMPANY ID = '*DOWN', or the GROUP TYPE CODE = 'SD', or the CUSTOMER TRANSACTION ID = '*STOP', then execute the following two programs, then end this program.

- - The first is to set the CPI-C Deal Type by executing 'CMSDT' program with the following parameters:

8 Character (Input) CONVERSATION ID
9/0 Binary (Input) CONVERSATION DEAL SET TYPE, loaded with the value 1.
9/0 Binary (Output) PROGRAM RETURN CODE

- - The second is to DeAllocate a CPI-C communication conversation by executing 'CMDEAL' program with the following parameters:

8 Character (Input) CONVERSATION ID
9/0 Binary (Output) PROGRAM RETURN CODE

- IF a non-shutdown data queue entry is received, then use its value to repetitively read each corresponding record in the ARMS Transmissions to be Sent (AMSEND) file. Load each retrieved record's ARMS format to a temporary DATA TO BE SENT array of 128 character elements and 68 elements, increment the RECORD COUNTER by 1 and the SENT DATA LENGTH by 128.

- - IF the RECORD COUNTER is greater than 69 (68 x 128 = 8704 characters, full communication buffer), or at least 1 record was read from the ARMS Transmissions to be Sent (AMSEND) file, then execute the 'CMSEND' program with the following parameters:

8 Character (Input) CONVERSATION ID
8704 Character (Input) DATA TO BE SENT ARRAY
9/0 Binary (Input) SENT DATA LENGTH (loaded with the current increments of 128 bytes per record).
9/0 Binary (Output) REQUESTOR ID, loaded with 0 (zero).
9/0 Binary (Output) PROGRAM RETURN CODE

- - Read from the loaded DATA TO BE SENT ARRAY, starting at the first element for every element loaded, write a record to the ARMS Transmissions Sent (AMSENDLOG) file and then send a data queue entry for these written records to the ARMS Time Line Input Data Queue (DQAM70V1).

- - Reset the SENT DATA LENGTH and the RECORD COUNTER to 0 (zero) and clear the DATA TO BE SENT ARRAY.

ARMS Process Report

@Files: (CRUD)

AMSEND (-R--)
AMSNDLOG (C---)

@Embedded Data/Constants:

'ADDLIBLE LIB(QSYS2)' for use in the execution of the program QCMDEXC to execute this command so that the IBM System Library for CPI-C programs' objects would be available for use.

'*LIBL' used as the input parameter for the LIBRARY NAME for the execution of the Receive Data Queue and Send Data Queue programs.

Process

Hierarchical numeric ID: 1.1.1.1.19.8
Coded name: EDMSLUA
Name: PGM Send X12 Transmission (EDMSLUA)
Comment: @Purpose:

To send any wrapped EDI X12 transactions that are ready to be sent.

@Operational Method:

- This program runs in an infinite loop from the time it is submitted until a shutdown flag is returned by the called program EDZSCWA. The program is submitted by the ETD: EC/EDI: ARMS Trading Partner Insurance Company Comm./Mapping Submitter (AM***COMP) program to run in the ARMS subsystem on the centralized ARMS host computer platform and is passed the following input parameters:

5 Character COMPANY PROFILE ID with transactions to be sent (such as '***01')
15 Character NETWORK ID, as defined in Any EDI integration software, for this ARMS Company Profile ID to be used for this network communication session. (such as 'ProfileName')
6 Character PORT ID, as defined in any EDI integration software, to be used for this network communication session (such as 'xxxxx').
10 Character SCRIPT ID, as defined in any EDI integration software, as the communication script to be used for this company in the ARMS connected network. (such as 'xxxxxS')
1 Character VERSION FLAG, Either '0' = Test or '1' = Production.

- Repetitively execute the ARMS ANY EDI INTEGRATION SOFTWARE Generic Send Transaction Notify Program (EDZSCWA) with the following parameters, until the *DOWN FLAG indicates a shutdown has been received:

5 Character (input) passed input parameter, COMPANY PROFILE ID with transactions to be sent (such as 'xxxxx')
6 Character (input) passed input parameter, PORT ID, as defined in any EDI integration software, to be used for this network communication session. (such as '***01')
1 Character (output) *DOWN FLAG shutdown data queue entry received. (Values are 'Y' = Yes or 'N' = No)
1 Character (output) TRANSACTIONS PROCESSED TO BE SENT FLAG, (Either 'Y' = Yes or 'N' = No)
1 Character (input) passed input parameter, VERSION FLAG (Either '0' = Test or '1' = Production)

ARMS Process Report

IF the returned TRANSACTIONS PROCESSED TO BLANK FLAG is Yes, then Send transactions to trading partner based on Company ID, Network, Port, and Communication's Script by execution of the any EDI integration software's command for API: Send connections ready to send - batch.

- Upon return of control from EDZSCWA, if the shutdown flag has been set, end the program.

- IF there is an error is due to communications error line ARMS-CONNECTED VALUE-ADDED NETWORK - (UEX0567), or,
IF there is an error is due to communication's failure, (UEX0573),
IF there is an error is due to no connections ready to send which match the selection parameters, (UEX0517), or,
IF there is an error while processing a connection, (UEX0519), or,
IF there is any other error escape message, then execute the error routine.

The error routine consists of: Output a spooled CL program dump; Notify ETD and ARMS On-Call personnel, end the program.

@Notes:

Program EDZSCWA waits on the data queue that indicates the presence of wrapped transactions and returns the indicator for the presence of these transactions to this program.

Process

Hierarchical numeric ID: 1.1.1.1.19.12
Coded name: EDLSIMA
Name: PGM Map Proprietary to X12 Transmission (EDLSIMA)
Comment: @Purpose:

To drive the translation of the ARMS proprietary transaction set into the ANSI X12 EDI transmission set.

@Operational Method:

IF translation errors are encountered or any unspecified escape error messages are generated, Dump the CL program, Call both the ETD PSSR and the ARMS PSSR programs, passing the current job's JOBNAME, 'EDLSIMA1', and a single blank character to notify ETD and ARMS On-Call staff. Then End the program.

Retrieve the EC/EDI Send Mapper Debug Flag 1 character data area to check the DEBUG FLAG value.

IF the passed input parameter field VERSION is 'T', then perform overrides to the "test" interface files for 272 set to the ELEDITST E's EDI Test Data Library for the following 272 transaction set files and their associated test interface save files: S272DTL1, S272DTP1, S272HDR1, S272ER1, S272K21, S272NAM1, and S272PER1. Additionally for the following 824 transaction set files: S824HDR1, S824ER1, S824OTI1, S824N1, S824NTE1, S824ME1, and S824PER1. Also, IF not on the "DEV" (development) platform, then override the following DDM (distributed data management) remote file on the "DEV" platform in the ELEDITST EDI Test Data Library for the ARMS Cross-Reference File (DAMXREFL1) and the ARMS Authorization Detail File (DAMAUTD).

Allocate the following output sender segment detail interface work files for exclusive use:

S272DTL1
 S272DTP1
 S272ER1
 S272HDR1
 S272K21
 S272NAM1
 S272PER1

S824HDR1
 S824ER1
 S824OTI1
 S824N1
 S824NTE1
 S824ME1
 S824PER1

Derive the Trading Partner (COMPANY ID) Send Communications Version 1 output data queue name. IF the passed VERSION is '1' (production), then build the data queue name by the concatenation of the first three (3) characters of the passed input parameter field COMPANY ID, the constant "SE" (to SEnd), and the passed input parameter field VERSION.

ELSE (is test), then build the data queue name by the concatenation of the characters from position 2 through 4 of the passed input parameter field COMPANY ID, the constant "SE" (to SEnd), and the passed input parameter field VERSION.

Perform the following until SHUTDOWN FLAG or an exception error occurs:

- Call the ARMS X12 Send Interface Version 2 program (AMZSIMA) with the following parameters:

1	Character	(Output) Shutdown Flag	(possible values: 'N'=No or 'Y'=Yes)
3/0	Decimal	(Output) Number of 272 Transaction Sets	
3/0	Decimal	(Output) Number of 824 Transaction Sets	
5	Character	(Input) Company ID, passed input parameter field value	

Go to this program

- IF transactions were written to the interface files (NUMBER OF 272/824 TRANSACTION SETS > 0), then execute the EDI integration software Create Application Data Batch (batch) command for the Property and Casualty Loss Notification Outbound Version 1 application ID for the production or test data processed, depending on the value of the input parameter VERSION. Then send a '*DATA' (meaning "data is available to be sent") 5-character data queue entry to the derived output Trading Partner (COMPANY ID) Send Communications Version 1 data queue.

- IF the retrieved data area's DEBUG FLAG is 'Y' (program is running in debug mode), the copy/add the transaction set segment detail interface work files to their associated save archive files.

- Clear these transaction set segment detail interface work files.

- IF the called ARMS X12 Send Interface Version 2 program (AMZSIMA) returned with SHUTDOWN FLAG parameter = 'Y', then send three (3) '*DOWN' 5-character data queue entries to the derived output Trading Partner (COMPANY ID) Send Communications Version 1 data queue and end this program.

- ELSE, repeat the above steps.

@Files: (CRUD)

ARMS Process Report

EDEXTDEBUG (-R--) 1 character data area : EC/EDI Send Mapper Debug
Flag (possible values: "N"=No or "Y"=Yes)

@Notes:

This program is maintained by the Emerging Technologies Department (ETD).

This program is submitted as a never-ending program batch job with the job name 'Lxxxxx1SM' (where "xxxxx" is the Trading Partner Company Profile ID) and specifies using the any EDI integration software JOBDP job description on the centralized ARMS application host computer system with the following input parameters:

1 Alphanumeric VERSION (possible values: "1"-production or "T"-testing)
5 Alphanumeric COMPANY ID

Process

Hierarchical numeric ID: 1.1.1.1.19.30

Coded name: AMZSIMA

Name: PGM Create Send Interface Work Files for Any Transactions to be Sent (AMZSIMA)

Comment: @Purpose:

To receive any outbound transmission to be sent and format it into the appropriate send transaction interface work files for mapping by the EDI integration software (any third-party software package).

@Operational Method:

- Set the delay time for gathering transactions from data area AMEXTSM. To improve performance, the EDI integration software is not invoked for each transaction. The delay time is the duration for which the program will wait to see if any more transactions are available to be sent before returning to the calling program.

- Receive data to be sent to insurance company in ARMS proprietary format and convert the format to send interface file format for X12 transaction sets 272 property and casualty loss notification and work assignments (rentals)} and 824 application advice designed to accommodate the business need for reporting the acceptance, rejection, or acceptance with change of any transaction set}.

- Use file EDPREF3 to determine which insurance company expects which X12 transaction set for a particular ARMS proprietary group type.

@Files: CRUD)

- AMSEND	(-R--)
- AMTRNLOG	(-R--)
- S272HDR1	(CRU-)
- S272NAM1	(CRU-)
- S272DTL1	(CRU-)
- S272DTP1	(CRU-)
- S272ER1	(CRU-)
- S272K21	(CRU-)
- S272PER1	(CRU-)
- AMSNDLOG	(-R--)
- EDPREF3	(-R--)
- S824HDR1	(CRU-)
- S824ER1	(CRU-)
- S824N1	(CRU-)

ARMS Process Report

S824OTI1	(CRU-)
S824PER1	(CRU-)
S824NTE1	(CRU-)
S824ME1	(CRU-)
AMAUTD	(-R--)
AMXREFL1	(-R--)
AMEXTSM	(-R--) Data Area

@Notes: State

Currently, this used for the ANSI X12 '272' Rental Management Transaction Set used for Trading Partner Company and for the ANSI X12 '824' Consolidated Payment Transaction Set used for a different Trading Partner Company.

Currently only one ARMS Trading Partner Company will be sending 824 transactions but CIECA standards will be used in the future by other insurance companies to process status type transactions such as: 'AC', 'RN', 'ER'.

Process

Hierarchical numeric ID: 1.1.1.1.20

Coded name:

Name: AUT Generate Authorization Maintenance in Behalf of R.M.T.P.

Comment: @Definition: This automated process allows for an Enterprise In-House person to perform Authorization Management Maintenance on behalf of a Rental Management Trading Partner. Currently this is restricted to the processing of Authorization Changes and Cancellations. Only one ARMS Trading Partner Company is using this currently.

Process

Hierarchical numeric ID: 1.1.1.1.20.1

Coded name: AACHNG

Name: PGM AADRVR Program to PASSTHRU to RARMS (AACHNG)

Comment: @Definition: This program is used to pass through to the Centralized ARMS machine. It is used by In-house Enterprise Rental management personnel to access an interactive screen for use in processing AT-C, and or CN's on behalf of an ARMS Trading Partner Company.

@Operational Method: All this program does is start pass-through and starts the AACHNGA program. End program.

Process

Hierarchical numeric ID: 1.1.1.1.20.2

Coded name: AACHNGA

Name: PGM Program to start AACHGR (AACHNGA)

Comment: @Definition: This program is used to facilitate In-house Enterprise Rental management personnel access to an interactive screen for use in processing AT-C, and or CN's on behalf of a ARMS Trading Partner Company.

@Operational Method: This program is started by AACHNG on the Centralized ARMS machine. This program will call program AACHGR and then when control is returned it will end Pass-Through returning the user to the distributed machine.

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.1.1.20.3

Coded name: AACHGR

Name: PGM ARMS Trading Partner Company Authorization Change (AACHGR)

Comment: @Definition: This program is used to allow In-house Enterprise Rental management personnel to access an interactive screen for use in processing AT-Changes, and/or Cancellations on behalf of a ARMS Trading Partner Company.

@Operational Method: The System Name is retrieved and the users authority to the program is validated. If the user is not authorized, *INLR is turned *ON and the program exits.

If the user is authorized, all screen values and indicators are reset and Screen 1 is displayed. F3-Exit will exit the program. F5-Next will reset all screen values and indicators and redisplay the screen. F10-Process will begin processing, if all edits pass, an authorization change or cancellation. F11-Cancel will display Screen 2, if all edits pass.

After the user enters a valid combination of Group, Branch, Reservation, and Ticket, the ARMS cross-reference record is retrieved and the current status is used to determine possible transaction types. Authorization change requests are only allowed when the cross-reference status is 'R', 'O', or 'I'. Cancellation requests are only allowed when the cross-reference status is either 'R' or 'I'.

If an authorization change request is being processed, the new values are compared with the old values to ensure a change is made and edited for validity. If edits pass, necessary formats (APPD01, ADJD01, AUTD01, SURD01, COMD01) are loaded for an AT-C. Records are written to AMSET. A transaction log entry is written to AM004P. Data queue entries are sent to DQAM25V1 and DQAM70V1. Screen 1 is then redisplayed.

If a cancellation request is being processed, necessary formats (APPD01, ADJD01, CAND01, COMD01) are loaded. Records are written to AMSET. A transaction log entry is written to AM004P. Data queue entries are sent to DQAM25V1 and DQAM70V1. Screen 1 is then redisplayed.

@Notes: This is a temporary program for use until the ARMS Trading Partner Company completes development of their system.

@Files: (CRUD)

- AMXREFL1 (-R--)
- AMXREFL2 (-R--)
- AMRNTD (-R--)
- AMADJD (-R--)
- AMAUTD (-R--)
- AMSURD (-R--)
- AMXBCO (-R--)
- CUSTMAST (-R--)
- AMSET (C--)
- AM004P01 (C--)

Process

Hierarchical numeric ID: 1.1.1.2

Coded name:

Name: BT Manage Customer Remittance

ARMS Process Report

Comment: @Definition: The set of automated activities that are used in management of remittances, this is the result of an inbound transmission received from the trading partner who is engaged in electronic invoicing with us.

@Purpose: To allow EDI Trading Partners to receive and manage billing invoices electronically.

Process

Hierarchical numeric ID: 1.1.1.3

Coded name:

Name: BT Manage Customer Message

Comment: @Definition: The set of automated activities that are used in management of customer messages, this is the result of an inbound transmission received from the trading partner who is engaged in electronic messaging with us.

@Purpose: To allow the Rental Management Trading Partner to send messages to the Rental location electronically without the need of using the telephone.

Process

Hierarchical numeric ID: 1.1.1.2

Coded name:

Name: AA Rental Systems Business Transactions

Comment: @Definition: This application area represents all EDI transmissions that are sent from any of the Enterprise Rent-A-Car current internal rental application systems to the Rental Management Trading Partner system for: authorization requests, notification, cancellations, and extension; Vendor messages; Vendor Invoices

Process

Hierarchical numeric ID: 1.1.1.2.1

Coded name:

Name: BT Request for Authorization Management by Vendor

Comment: @Definition: The set of automated activities that are used in the request for authorization and extension of a rental that is seen as an outbound transmission directed toward the trading partner.

@Purpose: To provide the rental offices an automated /electronic way to request authorizations and extensions for rentals from Rental Management Trading Partners without having to use the telephone.

Process

Hierarchical numeric ID: 1.1.1.2.2

Coded name:

Name: BT Generate & Send Rental Maintenance Notification

Comment: @Definition: The set of automated activities that are used in the creation of and transmission of rental notification seen as an outbound transmission directed toward the trading partner.

@Purpose: To notify the trading partner of the open or close of an authorized rental.

Process

Hierarchical numeric ID: 1.1.2.3

Coded name:

Name: BT Manage Vendor Messages

Comment: @Definition: The set of automated activities that are used in management of vendor messages, this is the result of an action on the part of a rental agent and results in a transmission sent to a trading partner that engages in electronic messaging with us.

@Purpose: To allow the rental office to send electronic messages to the rental trading partner without the need to pick up the telephone.

Process

Hierarchical numeric ID: 1.1.2.4

Coded name:

Name: BT Generate & Send Electronic Billing Invoice and Batch Extensions

Comment: @Definition: The set of automated activities that are used in the creation and transmission of electronic invoices to the trading partner.

@Purpose: To allow for paperless billing of rentals to customers that are set up for EDI billing.

Process

Hierarchical numeric ID: 1.1.2.4.1

Coded name:

Name: BAT Package and Send Transaction / Generate Vendor Extension Request

Comment: @Definition: Package information held on the Centralized ARMS System into a batch package and send the information to the rental management trading partner or the direct bill trading partner. Or on the distributed rental systems generate requests for authorization extensions (RE's) based upon specific conditions of rentals as determined from the Rental Database

@Purpose: To prepare and send outbound transactions to the EDI Trading Partners. And to insure that rental extensions are requested in a timely fashion.

@Notes: The information held could be invoices, office location updates, service area updates and/or location rate updates.

Process

Hierarchical numeric ID: 1.1.2.4.1.1

Coded name:

Name: BAT Package and Send Transactions to specified ARMS Trading Partner Companies

Comment: @Definition: Package information held on the Centralized ARMS System into a batch package and send the information to specified ARMS Trading Partner Companies other than ANSI X12.

@Purpose: To prepare and send outbound transactions to the specified ARMS Trading Partner Companies other than ANSI X12 EDI Trading Partners.

@Notes: The information held could be invoices, office location updates, service area updates and/or location rate updates.

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.2.4.1.1.5

Coded name:

Name: DTQ Connect-Specific Communications Sender Input (DQxxxxxx)

Comment: @Definition: DQxxxxxx Connect Specific Communications Sender data queues are used to provide input to the programs that send transmissions to our EDI Trading Partners.

Process

Hierarchical numeric ID: 1.1.2.4.1.1.15

Coded name: AM0150

Name: PGM Package Batch Transaction (AM0150)

Comment: @Purpose:

To package control start records before and control end records after each Vendor Invoice (IN) and each Vendor Extension Request (RE) data set, to form a complete transmission set. Control start and end records are used to route the transmission and indicates the type of transaction set being sent to the Trading Partner.

@Operational Method:

- For each transaction data set
- Build the control start record format (Transmission, Group, & Set) and write record formats to send file (AMSEND)
- For each transaction data set record format
- Translate the location (AUTD01/ELCD01/IEBH01
- IF record format NE to Insured Detail (INSD01)
Write record format to send file (AMSEND)
- IF transaction data set EQUALS Vendor Invoice (IN)
Write record format to ARMS Billing Log
- Build the control end record format (Set, Group, & Transmission) and write record formats to send file (AMSEND)
- Sends a data queue to the connect-specific communications sender

@Notes:

- Control start/end records must match:
- Transmission start
Group Start
Set Start
:
Set End
Group End
Transmission End

@Files (CRUD)

L811C (-R--)
ARMSPR3 (-R--)
AMSEND (C---)
AMBILLOG (C---)

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.2.4.1.2

Coded name:

Name: BAT Package and Send Transactions for X12

Comment: @Definition: Package information held on the Centralized ARMS System into a batch package and send the information to the X.12 trading partner.

@Purpose: To prepare and send outbound transactions to the X-12 EDI Trading Partners.

Process

Hierarchical numeric ID: 1.1.2.4.1.2.1

Coded name: UT_999M

Name: PGM Generic ILE Error Handler (ErrHandler)

Comment: @Purpose:

To handle exceptions for application programs, by determining the proper response reacting in a predetermined way. These responses include sending a MS01 message and sending pages to the appropriate personnel.

@Operational Method:

Applies To: (ILE) Toolbox Error Handling

This generic application program error handler may be executed by an OPM (Original Program Model) program by executing an intermediate ILE RPG wrapper program, "UT_999M" - ISS:IS Global Error Handling Module (OPM Callable). This program was created with the bound service program "ISLIB/UT_H05S" - ISS: Universal Toolbox (*CALLER) that was created with the bound module "ISLIB/UT_H05S012" that perform the necessary service functions. This program is executed with the following parameters:

Parameter	Type-Usage	Size	Description
programID	Char - Input	10	Program issuing the
errorID	Char - Input	7	Error ID
msgDta	Char- Input	75	Substitution
data for message description			
Return Code	Char - Output	1	Returns 1 for
failure, 0 for success			

Before using this function/procedure, three files must be updated with the necessary information. These files are UT991P, UT992P, and UT993P.

The existing entries must be used as examples to follow.

Syntax:

rc = ErrHandler(<programID> :<errorID> [:<msgDta>] [:<errorNoAction>])

Elements:

Argument	Type	Size	Description
rc (Return Code)	Char	1	Returns 1 for failure, 0 for
programID	Char	10	Program issuing the error

ARMS Process Report

errorID Char Error ID found in the
 UT991P file for the associated Program ID
 msgDta Char 32767 Substitution data for message
 description
 errorNoAction Char 1 Parameter for future update,
 currently has no function

@Notes:

This ILE RPG wrapper program is currently executed only by the AMZ811A program.

The following necessary records were placed into the UT991P ISS:IS Error Message ID File in order to execute the error handling functions properly:

Program Issuing Error	Message Error ID	Message File	Name	Description ID
AMZ811A	NORECFD	AMMSGF	AMM0003	
AMZ811A	SEQERR	AMMSGF	AMM0001	
AMZ811A	INVCMP	AMMSGF	AMM0002	
AMZ811A	*PSSR	AMMSGF	AMM0004	

The following necessary records were placed into the UT992P ISS:IS Error Message Response File in order to execute the error handling response functions properly:

Program Issuing Error	Error ID	Sequence ID	Code	Error Response
AMZ811A	*PSSR	1		ARMSPAGE
AMZ811A	*PSSR	2		ARMSMS01
AMZ811A	INVCMP	1		ARMSPAGE
AMZ811A	INVCMP	2		ARMSMS01
AMZ811A	NORECFD	1		ARMSPAGE
AMZ811A	NORECFD	2		ARMSMS01
AMZ811A	SEQERR	1		ARMSPAGE
AMZ811A	SEQERR	2		ARMSMS01

The following necessary records were placed into the UT993P ISS:IS Response/Program File in order to specify the execution of the error handling response mode functions:

Response Code	Exit Program
ARMSPAGE	UT_999M005
ARMSMS01	UT_999M006

Process

Hierarchical numeric ID: 1.1.2.4.1.2.19
 Coded name: AMZ832A
 Name: PGM Send Interface X12 Transaction Set 832 (AMZ832A)
 Comment: @Purpose:

ARMS Process Report

To convert Enterprise proprietary EDI formats held on the Centralized ARMS System into X.12 standard formats (transaction set 832) which will then be sent to the X.12 trading partner

@Operational Method:

- IF transaction have been written to application database (AMSNDLOG), clear application work files.
- Attempt to receive the next DQ832S data queue entry with a key of Company Id.
- IF a data queue entry was not received, close program files (if applicable) and end program
- IF a data queue entry was received THEN
 - IF no records were retrieved from the application database send file(AMSEND), using the received data queue entry - page the on-call staff and get next data queue entry.
 - IF records were retrieved from the application database send file (AMSEND), using the received data queue entry THEN
 - IF the transaction is in the correct sequence, notify the on-call staff
 - Use the Rental Management Trading Partner electronic commerce profile to determine if they receive X12 standard format - 832 for this specific application group type. IF not exit the program.
 - For each invoice, attempt to load the EDI integration software work files:
 - Retrieve records from application database
 - IF the invoice charge code or unit of measure is not found, skip this invoice and notify ARMS on-call.
 - Load the 832 extol work file and writing them to the send log file (AMSNDLOG).
 - Update 832 header work file.
 - Send a data queue entry to timeline (DQAM70)

@Files:	(CRUD)
AMSEND	(-R--)
AMSNDLOG	(CR-D)
AM811P01	(-R--)
S832HDR1	(CR-D)
S832LIN	(CR-D)
S832PER	(CR-D)
S832CTP	(CR-D)
S832N1	(CR-D)
EDPREF3	(-R--)
AM832ERR	(C---)

Process

Hierarchical numeric ID: 1.1.2.4.1.2.20

Coded name:

Name: DTQ Input for X12 Transaction Set 832 (DQ832S)

Comment: @Definition: DQ832S is a data queue used to provide input for X.12 transaction set 832 to PGM AMZ832A.

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.2.4.1.2.23
Coded name: AMZ811A
Name: PGM Send Interface X12 Transaction Set 811 (AMZ811A)
Comment: @Purpose:

To convert Enterprise proprietary EDI formats held on the centralized ARMS application system host platform into X12 standard format (transaction set 811) which will then be sent to the X12 trading partner. .

@Operational Method:

- IF the transaction set has been written to the application database for Sent Transaction Log (AMSNDLOG) file, then clear the application work files.

- Attempt to receive the next DQ811S data queue entry with a key of Trading Partner Company ID.

- IF a data queue entry was not received, close program files (if applicable) and end program

- IF a data queue entry was received THEN

- IF no records were retrieved from the application database send file(AMSEND), using the received data queue entry - page the on-call staff and get next data queue entry.

- IF records were retrieved from the application database send (AMSEND) file, using the received data queue entry, THEN:

- IF the transaction is in the correct sequence, notify the on-

Call staff

- Use the Rental Management Trading Partner electronic commerce profile to determine if they receive X12 standard format - 811 for this specific application group type. IF not, then exit the program.

- For each invoice, attempt to load the Extol work files:

-- Retrieve records from application database

-- IF the invoice charge code or unit of measure is not found, skip this invoice and notify ARMS on-call.

-- Load the 811 Extol work file and writing them to the send log

(AMSNDLOG) file.

-- Update 811 header work file.

-- Send a data queue entry to timeline (DQAM70)

@Notes:

- This program run with in CLL811 between 3:00am and 5:00am Central time zone.

- This program only processes invoice transmission sets.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer).

@Files: (CRUD)

AMSEND (-R--)

AMSNDLOG (CR-D)

AM801P01 (-R--)

AM802P01 (-R--)

AM811P01 (-R--)

ARMS Process Report

ARMSPR1	(-R--)
AMAUTD	(-R--)
S811HDR1	(CR-D)
S811NAM1	(CR-D)
S811DTL1	(CR-D)
S811IT1	(CR-D)
S811VEH1	(CR-D)
EDPREF3	(-R--)
AM811ERR	(C--)

Process

Hierarchical numeric ID: 1.1.2.4.1.2.24
Coded name:
Name: DTQ Input for X12 Transaction Set 811 (DQ811S)
Comment: @Definition: DQ811S is a data queue used to provide input for X.12 transaction set 811 to PGM AMZ811A.

Process

Hierarchical numeric ID: 1.1.2.4.1.2.26
Coded name: EDMSOFA
Name: PGM Create Extol Envelope for Office Data (EDMSOFA)
Comment: @Purpose:
To create the transmission envelope for rental location/rates maintenance transaction sets for the sender interface to the Extol EDI Translation Software.
@Operational Method:
- Call this program after the interface program AMZ832A has already put the data in AMSEND. This program will create the necessary EDI transmission envelope and transmit it to the insurance company.
@Notes: Set Environment as ADDLIBLE ELEDIxxx where "xxx" is the first 3 characters of the TRADING PARTNER PROFILE ID.
This program is executed only on the centralized ARMS host application system platform.
This program is currently stewarded by the Emerging Technologies Department's (ETD) Electronic Data Interchange (EDI) projects team.

Process

Hierarchical numeric ID: 1.1.2.4.1.2.27
Coded name: EDMSINA
Name: PGM Create Extol Envelope for Invoice Data (EDMSINA)
Comment: @Purpose:
To create the transmission envelope for rental billing invoice transaction sets for the sender interface to the Extol EDI Translation Software.
@Operational Method:
- Call this program after the interface program AMZ811A has already put the data in AMSEND. This program will create the necessary EDI transmission envelope and transmit it to the insurance company.

ARMS Process Report

@Notes: Set Environment as ADDLIBLE ELEDIxxx where "xxx" is the first 3 characters of TRADING PARTNER PROFILE ID.

This program is executed only on the centralized ARMS host application system platform and is executed from the ARMS Packaging Translation Job (CLL811) program.

This program is currently stewarded by the Emerging Technologies Department's (ETD) Electronic Data Interchange (EDI) projects team.

Process

Hierarchical numeric ID: 1.1.2.4.1.2.28

Coded name:

Name: DTQ Input for X12 Sender (DQxxxSC1) "xxx" = Company

Comment: @Definition: DQxxxSC is a data queue used to provide input for X.12 transaction sender (xxx = Company).

Process

Hierarchical numeric ID: 1.1.2.4.1.3

Coded name:

Name: BAT Package and Send Transactions to specified ARMS Trading Partner Companies

Comment: @Definition: Package information held on the Centralized ARMS System into batch package and send the information to specified ARMS Trading Partner Companies.

@Purpose: To prepare and send outbound transactions to specified ARMS Trading Partner Companies.

Process

Hierarchical numeric ID: 1.1.2.4.1.3.7

Coded name: AM***SNB

Name: PGM Send Batch Transmission (via ARMS-connected Value-Added Network AM***SNB)

Comment: @Purpose:

To send from ARMS to specified ARMS Trading Partner Companies a Batch EDI Transmission of proprietary transaction data sets via ARMS-connected Value-Added Network. The transaction set functional group types sent are rental billing invoice documents ("IN") and rental location/rates maintenance documents ("OF").

@Operational Method:

Clear the EDI integration software Local Data Area data structure.

Retrieve the EDI integration software User Profile File record for ACCOUNT = 'ERAC' and USERID = 'ERACARM'.

Load the entire retrieved record value to the EDI integration software Local Data Area data structure's subfield for PROFILE RECORD.

Load the remainder of the EDI integration software Local Data Area data structure's subfields as follows:

'IEERACARM '	User files library (E4LBLB)
'IESTRM '	Stream file name (E4LBST)
'IERCVD '	Received file name (E4LBRC)

ARMS Process Report

'IEMSGS'	Messages file name (E4LBFL)
'IEFILE'	Received files file name (E4LBFL)
'IESLOG'	Session log file name (E4LBFL)
'0'	Base submit flag (E4LSBF)
'0'	Base calling code (E4LBCL)

IF the passed input COMPANY PROFILE ID parameter is a specified ARMS Trading Partner Companies production ('***01'), load the COMMAND FILE NAME with 'IE***SND'.

IF the passed input COMPANY PROFILE ID parameter is a specified ARMS Trading Partner Companies testing ('T***T'), load the COMMAND FILE NAME with 'IE***SNDT'.

Output to this program's Local Data Area (to be used by the EDI integration software) from the loaded EDI integration software Local Data Area data structure.

Execute the EDI integration software base ('IEIFEXEC') program to execute the receive file according to the information just loaded in the current batch job's Local Data Area.

Retrieve the EDI integration software Local Data Area and load into the EDI integration software data structure.

IF the EDI integration software Local Data Area data structure's Base Return Code is not a normal completion ('00'), then execute the Program Status Subroutine (*PSSR) to produce a formatted RPG dump and execute Perform Internal Error Paging and Messaging ('AMPSSR') program.

Retrieve the current time and date and load into the appropriate Send Log File record fields.

Position the ARMS specified ARMS Trading Partner Companies Send Batch File at beginning of file and then read each record and perform the following until end of file:

..IF the retrieved record's Record Format ID is Transmission Start (TSMS01), move the retrieved record to the Transmission Start record format's data structure. Move the Transmission Start record format's TRADING PARTNER ID and TRANSMISSION CONTROL ID to the appropriate Send Log File fields and to the appropriate ARMS Time Line Data Queue data structure subfields.

..Write each retrieved record to the ARMS Send Log File.

..IF the retrieved record's Record Format ID is Transmission End (TSME01), then send a data queue entry to the ARMS Time Line data queue (DQAM70V1) that is associated with the send details written out to the ARMS Send Log.

@Notes:

This program is executed by the CLL811 CL batch program and is executed on the centralized ARMS host system platform ("RARMS") in the ARMS subsystem. It is passed a single input parameter of a 5-character COMPANY PROFILE ID.

@Files: (CRUD)

AM***SNB (-R--) ARMS Transactions to be Sent to specified ARMS Trading Partner Companies via Send Batch

ARMS Process Report

IEPROFL (-R--) EDI integration software User Profile File (in
the IBM EDI integration software Production Library for *USER)
AMSNDLOG (C---) ARMS Transmissions Sent Log File

Files used as externally-defined data structures:

External Definition
EX40LD EDS ETD: EDI integration software Local Data Area
TSMS01V1 EDS Transmission Start Record Format
DQAM70V1 EDS ARMS Time Line Input Data Queue external description

@Embedded Data/Constants:

'AM***SNB' PROGRAM ID

Used as output to the job's Local Data Area for the execution of the EDI
integration software base ('IEIFEXEC') program to execute the receive file:

'IEERACARM ' User files library
'IESTRM ' Stream file name
'IERCVD ' Received file name
'IEMSGS ' Messages file name
'IEFILE ' Received files file name
'IESLOG ' Session log file name
'0' Base submit flag
'0' Base calling code
'IE***SND a ARMS Trading Partner Companies production ('***01')
COMMAND FILE NAME
'IE***SNDT a ARMS Trading Partner Companies testing ('T***T')
COMMAND FILE NAME

@Improvement Opportunities:

1.) Convert from OPM RPG to ILE RPG program.

2.) Replace the constant 'AM***SNB' with the Program Status Data
Structure subfield PROGRAM ID.

Process

Hierarchical numeric ID: 1.1.2.4.1.3.10
Coded name: AM0125
Name: PGM Create Customer Batch Time Work File (AM0125)
Comment: @Purpose:

To load a temporary job work file with each of the financial management
or rental management trading partners preferred batch packaging/sending start times
for production and test transaction data sets.

@Operational Method:

- Using the billing times from the ARMS profile and whether a company
receives electronic bills, create the billing times work file for each company id in
the ARMS profile (ARMSPR3). If the company is on a VAN (Value Added Network), get
the billing time from the corresponding VAN profile.

@Files:

ARMS Process Report

ARMSPR1 (---)

ARMSPR3 (-R--)

L811D (C---) ARMS Company Profile IDs and Batch

Packaging/Sending Time preferences job work file, unsorted.

@Notes:

The following Trading Partner (Company) Profile ID values are specific ARMS communication connections that multiple trading partner utilize to exchange information with ARMS:

VN101 - an ARMS-connected VAN Information Systems Value-Added Network (VAN) software

VN201 - PC-ARMS

VN301 - ARMS/400

This is executed by the ARMS Batch Packaging/Send Job (CLL811) program on the centralized ARMS application system platform.

The ARMS system also enables electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer). Therefore, ARMS will receive a data queue entry to generate an electronic invoice (and create an associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) even though the trading partner never sent to ERAC (via ARMS) any initial rental authorization.

@Embedded Data/Constants:

'AM0125' as PROGRAM ID input parameter for execution of the ARMS Program Status Subroutine Exception Handler (AMPSSR) program.

@Improvement Opportunities:

1.) Convert this OPM RPG to ILE RPG.

2.) Replace the constant 'AM0125' with the Program Status Data Structure's subfield defined for PROGRAM ID.

Process

Hierarchical numeric ID: 1.1.2.4.1.3.13

Coded name: AM0140

Name: PGM Create Batch Transactions Work File (AM0140)

Comment: @Purpose:

To copy batch transaction data sets to a job file for batch packaging and sending for a specified trading partner.

@Operational Method:

- Write the *EOBD (End of Batch Data) APPD02 blank data record format to the AMPACK file for the current input parameter TRADING PARTNER PROFILE ID as the COMPANY PROFILE ID portion of the key and the '*EOBD' as the key's CUSTOMER TRANSACTION ID.

- Retrieve all the data ready to be packaged and sent for a specified insurance company (until the End of Batch Data) is retrieved, written to a job work file, except the End of Batch Data record.

ARMS Process Report

@Notes: This is executed by the ARMS Batch Packaging/Send Job (CLL811) program on the centralized ARMS application system platform with one 5-character input TRADING PARTNER PROFILE ID.

@Files:

AMPACK (CR--) Transaction Data Sets to be packaged for sending
L811B (C--) Batch Packaging Job Work File
ARMSKEY (-R--) EDS
APPD01V1 (-R--) EDS

Process

Hierarchical numeric ID: 1.1.2.4.1.3.14

Coded name: AM0151

Name: PGM Package Batch Transaction for specified proprietary and ANSI X12 ARMS EDI Trading Partner Companies (AM0151)

Comment: @Purpose:

To package control start records before and control end records after Vendor Invoices (IN) and Vendor Extension Request (RE) data set records, to form a complete transmission set. Control start and end records are used to route the transmission and indicates the type of transaction set being sent.

@Operational Method:

- For each transaction data set
--- Build the control start record formats (Transmission, Group, & Set)
--- IF company ID EQUALS '***01'
Write record formats to specified ARMS Trading Partner

Companies send Batch

(AM***SNB)
ELSE
Write record formats to send file (AMSEND)
--- For each transaction data set record format
----- Translate the location (AUTD01/ELCD01/IEBH01)
----- IF company ID EQUALS '***01'
Write record format to ARMS Trading Partner Companies send

Batch

(AM***SNB)
ELSE
Write record format to send file (AMSEND)
----- IF transaction data set EQAUL Vendor Invoice (IN)
Write record format to ARMS Billing Log
--- Build the control end record formats (Set, Group, & Transmission)
--- IF company ID EQUALS '***01'
Write record formats to an ARMS Trading Partner Companies send

Batch

(AM***SNB)
ELSE
Write record formats to send file (AMSEND)
--- Sends a data queue to the connect-specific communications sender

@Notes:

ARMS Process Report

```
- Control Start/end records must match:
--- Transmission start
    Group Start (IN)
        Set Start
        :
        Set End
        Set Start
        :
        Set End
    Group End(IN)
Transmission End
```

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer). Therefore, ARMS will receive a data queue entry to generate an electronic invoice (and create an associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) when the trading partner never sent to ERAC (via ARMS) any initial rental authorization.

```
@Files          (CRUD)

L811C            (-R--)
AM***SNB        (C---)
AMSEND          (C---)
AMBILLOG        (C---)
```

Process

Hierarchical numeric ID: 1.1.2.4.1.4

Coded name:

Name: BAT Generate Vendor Request for Authorization Extension

Comment: @Definition: To generate requests for authorization extensions (RE's) based upon specific conditions of the rentals as determined from the Rental Database. This process is triggered by TIME.

@Purpose: To insure that rental extensions are requested in a timely fashion independantly of the manual call back process of the branch without the need of calling a bodyshop or adjustor, for those Trading Partners that allow the receipt of RE's.

Process

Hierarchical numeric ID: 1.1.2.4.1.4.8

Coded name: CLL811

Name: PGM Transmit Batch Transactions (CLL811)

Comment: @Purpose:

To perform certain requests/services for our trading partners base on the host platform.

When on the Centralized ARMS system, to package any held information(invoices, office location) into batches and send the transmission to the trading partner.

When on the Distributed Rental Management system, to generate requests for authorization extension.

ARMS Process Report

@Operational Method:

- IF the host platform is the distributed rental system (includes claims connection system on CENTRAL), execute the procedure for generating authorization extension request(s).

- IF the host platform is the centralized ARMS system, batch package and send transmission to the trading partner.

1.) Use the Rental Management Trading Partner Profile - billing time, to initiate the batch package process at the predetermined time.

2.) At each scheduled time batch package any held information:

a) Use the Rental Management Trading Partner Profile to set the Trading Partner Company Profile ID.

b) IF Company ID flagged for production, retrieve company specific records from application database(OPNQRYF).

c) IF Company ID flagged as test

- IF DEV host platform not available
Send error message

ELSE

Retrieve company specific records from application Database on DEV host platform.

ENDIF

d) Copy Vendor invoices(IN) and office information (OF) to job file and insert 'bookmark' in application database file.

e) For each company id determine if there are records to package and how to package them:

- Single invoice per transmission (ANY ARMS-CONNECTED VAN)
- Multiple invoices per transmission
- IF Company ID = ARMS Trading Partner Company then EDI

Format is proprietary

ELSE

Convert proprietary format to X12

ENDIF

3.) Reorganize the EDI integration software User Profile file when the file contains at least 10% of deleted records.

@Files: (CRUD)

L811E (CR-D)

L811D (-R-D)

L811C (C--D)

L811B (C--D)

AMPACK (-R--)

AM***SNB (-R-D)

AM***SNBSV (C--)

Process

Hierarchical numeric ID: 1.1.2.4.1.4.9

Coded name: CC0BREV1

Name: PGM Send Claims Connection Vendor Extension Requests (CC0BREV1)

ARMS Process Report

Comment: @Purpose:

For those Rental Management Trading Partners that want this service, to interrogate the extension control file and determine if an authorization extension is needed. A data queue entry will be created for all contracts that have expired.

@Operational Method:

- For each extension control record
 - Verify the contract is authorized with the Rental Management Trading Partner having bill-to responsibility.
 - Determine if the extension / Due Back date has expired and the Rental Management Trading Partner hasn't sent a termination date.
 - Verify the Rental Management Trading Partner allows / is setup for this service - Generate request for extension.

@Files (CRUD)

CCEXTCTL (-R--)

CCMASTER (-R--)

REQSTCTL (-R--)

@Notes:

- Batch jobs runs during the startup of ARMS (L811) between 3:00am and 5:00am cst.
- This program only runs on the centralized financial system(Central).

Process

Hierarchical numeric ID: 1.1.2.4.1.4.16

Coded name: ECOBREV1

Name: PGM Send ECARS Vendor Extension Requests (ECOBREV1)

Comment: @Purpose:

For those Rental Management Trading Partners that want this service, to interrogate the callback control file and determine if an authorization extension is needed. A data queue entry will be created for all contracts that have expired.

@Operational Method:

- For each callback control record
 - Verify the callback is an authorized open ticket with the Rental Management Trading Partner having bill-to responsibility.
 - Determine if the extension date has expired and the Rental Management Trading Partner hasn't sent a termination date.
 - Verify the Rental Management Trading Partner allows / is setup for this service - Generate request for extension.

@Files (CRUD)

CB007P00 (-R--)

RACSMAS (-R--)

ECEXTCTL (-R--)

RACSUSP (-R--)

@Notes:

- Batch jobs runs during the startup of ARMS (L811) between 3:00am and 5:00am cst.

Process

Hierarchical numeric ID: 1.1.2.4.2

Coded name:

Name: AUT Generate CC Rental Billing Invoice

Comment: @Definition: The automatic process of generating billing invoices for rentals from third parties via CLAIMS CONNECTION.

Process

Hierarchical numeric ID: 1.1.2.4.2.1

Coded name: CC00INV1

Name: PGM Generate Electronic Invoice Detail (CC00INV1)

Comment: @Purpose:

To generate Vendor Invoice(IN) data set for sending to the Rental Management Trading Partner in response to the Claims Connection System closing the rental contract.

@Operational Method

```

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of
INC'.

- IF the receive data queue entry is a shutdown request, close program
files and send that request to data queue DQANDST to inform the distributed send
program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN
  - Rental contract meets authorization criteria
    - IF the necessary data base records are retrieved THEN
      - Build the applicable record format(s) and write the
        record
        to the distribution transaction file(ANDIST). Only
        build
        RATD01 record format if trading partner profile
        indicates it.
    ENDIF
  - Send a data queue entry on to the Distributed Send to
    Centralized data queue (DQANDST).
  ENDIF
- Send a data queue entry to Dispatch Data Queue
(DQAM60V1) with a key of 'DIS' to finish the request.
ENDIF

```

@Constants/Embedded Data:

```

- DAILY RENTAL - constant
- SALES TAX      - constant
- MISC EXP       - constant
- SURCHARGE     - constant
- 43-0724835 (tx id)- constant

```

@Improvement Opportunity

- Use program status data structure when referring to program name.

```

@Files          (CRUD)
CCCLSC          (-R--)

```


ARMS Process Report

CCSURDTL (-R--)
ANDIST (C---)
OFFDRB (-R--)

@Notes

- This program only runs on the centralized financial host platform.
- Application Data Set Formats
 - APPD01 - Enterprise internal format
 - IEBH01 - Invoice header format
 - RATD01 - Rate detail format
 - RNTD01 - Renter detail 1 format
 - REND01 - Rental Notification format
 - INSD01 - Insured format
 - IEBD01 - Invoice detail format
 - IEBT01 - Invoice total format
 - ELCD01 - Arms vendor location

Process

Hierarchical numeric ID: 1.1.2.4.3

Coded name:

Name: AUT Edit Outbound Transaction

Comment: @Definition: The automatic process of validating or editing the invoice formats and also updating the distributed ARMS database.

Process

Hierarchical numeric ID: 1.1.2.4.3.1

Coded name: AM0062V1

Name: PGM Edit Outbound Transaction (AM0062V1)

Comment: @Purpose:

To perform the distributed edit processing on any generated proprietary EDI transaction sets in the ARMS Distributed Transactions Sets to be Centralized (ANDIST) file prior to being sent to the centralized ARMS application host platform for routing.

@Operational Method:

This program endlessly receives the non-keyed input data queue entries from the input data queue (DQAM62V1) that the ARMS Dispatch Rental Systems Request program (AM0060V1) generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQANDST data queue and end this program.

For any non-shutdown data queue entry, check the non-shutdown key Group Type Code to be one of the following values:

'AC' (Rental Transaction Authorization Confirmation)
'ER' (Rejected Transaction in Error)
'IN' (Completed Rental Billing Invoice)
'OF' (Rental Office Location/Rates Maintenance)
'RA' (Rental Transaction Request for Authorization)
'RC' (Rental Transaction Request for Cancellation of Authorization)
'RE' (Rental Transaction Request for Authorization Extension)
'RN' (Rental Transaction Rental Notification Start/End)

IF the data queue entry's Group Type Code is NOT one of the above, then error the transaction set with the ERROR CODE = '16' ("Invalid Group Type Code").

Read each of the input data queue entry's associated Transaction Sets File records (currently only AMINTR is processed).

IF any record format read is not valid for the validated Group Type, then error the transaction set with ERROR CODE = '70' ("Invalid Format Specified in Set")

IF any following Record Format ID's are read, check:

..IF its associated record format data structure has already been loaded, then error the transaction set with ERROR CODE = '71' ("Only One Format of This Type Allowed in Set"):

'ADJD01'
'ADJD02'
'APPD01'
'AUTD01'
'CAND01'
'ELCD01'
'ERRD01'
'IEBH01'
'IEBT01'
'INSD01'
'OFFD01'
'REND01'
'RNTD01'
'RNTD02'
'RPRD01'
'VEDD01'

..ELSE, load the read record format into the record format's data structure.

..IF more than two (2) 'RATD01' record formats are read, then error the transaction set with ERROR CODE = '90' ("Invalid Format - Not Agreed Upon").

..Only allow loading and output of the following maximum occurrences of these Record Format ID's per transaction set, however, do not error the transaction if more are found, just do not allow loading, editing and output of them:

10 'COMD01' Comment Detail
35 'IEBD01' Invoice Detail
30 'SURD01' Approved Surcharge Detail

IF GROUP TYPE CODE = 'IN' (Invoice), then do the following:

..Validate that the required formats for this Group Type Code exist ('APPD01', 'ELCD01', 'IEBH01'; 'IEBT01'; 'INSD01'; 'REND01'; 'RNTD01'; at least one 'IEBD01'), and IF the 'IEBT01' record format's TOTAL AMOUNT DUE (IETDUE) is not zero. ELSE, error the transaction set with ERROR CODE = '69' ("Required Format Missing from Set").

..Perform transaction set group type specific record format's data element (field) value edits.

..IF no errors are found, then:

ARMS Process Report

....IF the data queue entry's VENDOR TRANSACTION ID is blank, then do the following (This is for any Financial Management Trading Partner who is able to receive any rental transaction's EDI Invoices but is not a Rental Management Trading Partner - Non-ARMS participating customer) :

.....Add a new ARMS Rental Transaction Cross-Reference File record by executing the ARMS Assign Next Vendor Transaction ID ('AMRTVSQ') program to retrieve the next VENDOR TRANSACTION ID value

.....Execute the ARMS Inquire/Update Rental Cross-Reference Record (AM1010V1) program for update to add the record passing this rental contract (ticket) identifier, rental location, the retrieve VENDOR TRANSACTION ID and a CUSTOMER TRANSACTION ID comprised of the current date in ccyyymmdd format.

.....IF its PROGRAM RETURN CODE is not blank, then error the transaction set with ERROR CODE = '82' ("Request Rejected - Cannot Identify Transaction ID").

.....ELSE, Update each loaded record format's key values data elements with the new CUSTOMER TRANSACTION ID and VENDOR TRANSACTION ID values.

.....ENDIF

....ENDIF

....Execute the ARMS Inquire/Update Rental Cross-Reference Record (AM1010V1) program for inquiry to check its STATUS CODE value.

....IF its PROGRAM RETURN CODE is not blank, then error the transaction set with ERROR CODE = '82' ("Request Rejected - Cannot Identify Transaction ID")

....ELSE, its PROGRAM RETURN CODE is blank, then

.....IF the STATUS CODE is not currently 'B' (Billed), 'A' (Audited), 'P' (Paid), 'Q' (Invoice Rejected by Customer), then Execute the ARMS Inquire/Update Rental Cross-Reference Record (AM1010V1) program for inquiry to update its STATUS CODE value to 'B' (Billed)

.....ENDIF

....ENDIF

....Write the record formats read to the ARMS Distributed Generated Transaction Sets to be sent to the centralized ARMS system for routing file (ANDIST)

....Send the input data queue entry as an output data queue entry to data queue (DQANDST) for subsequent transmission to the centralized ARMS host platform/machine.

..ENDIF

@Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters to execute only on the ECARS host system platform (NOT "CENTRAL" nor "RARMS"). Currently, the only (non-shutdown) transaction set group type code processed in this program is the proprietary EDI Invoice Transaction Set Invoice ("IN") group types. These invoice transaction sets were generated within a non-ARMS stewarded program, Generate ECARS Daily (Nightly) Billing batch (RB0010) program.

This is currently not performed on the Claims Connection host platform ("CENTRAL") because the Claims Connection process for Closing a Third-Party Rental Ticket sends a data queue entry to the ARMS Dispatch Rental Systems Request (AM0060V1) program to dispatch a data queue entry to the ARMS Generate Proprietary EDI Electronic Invoice Detail Transaction Set (CC00INV1) program, stewarded by ARMS.

IF any error is found, then do not process this transaction any further and execute the ARMS Handle Internal Error (AM0097V1) program, passing the input parameters of this data queue entry's key data elements and the currently loaded

ARMS Process Report

ERROR CODE value. This will notify ARMS On-Call of the error transaction set's data queue entry value and the applicable ERROR CODE. Do NOT output the transaction set's record formats to the ANDIST file nor send its data queue entry to DQANDST.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer) and AAA of California. Therefore, ARMS will receive a data queue entry to generate an electronic invoice (and create an associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) when the trading partner never sent to ERAC (via ARMS) any initial rental authorization.

@Files: (CRUD)

ANDQINFO (-R--) ARMS Data Queue Information File

AMINTR (-R--) ARMS Unedited Invoice Transaction Sets File

ANDIST (CR--) ARMS Distributed Generated Transaction Sets to be sent to the centralized ARMS system for routing.

@Embedded Data/Constants:

'0123456789'

'0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ '

'AM0062V1'

'AM0062V1 '

@Improvement Opportunities:

1.) Convert this OPM RPG program to an ILE RPG program.

2.) Delete the 'AM0062V1' program name constants and specify/use the Program Status Data Structure's PROGRAM ID data element.

3.) Delete the constants for valid numbers and use the TESTN operation for testing numerics.

4.) Add the submission of this program for execution on the Claims Connection host platform ("CENTRAL") because the mandatory field data values loaded from its rental system database must also be performed. The transaction set generation and editing processes should only be different in the fact that they access different rental systems databases.

5.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.

Process

Hierarchical numeric ID: 1.1.2.4.3.4

Coded name:

Name: DTQ Input to AM0062 (DQAM62V1)

Comment: @Definition: DQAM62V1 is a data queue used to provide input for PGM AM0062V1 which processes invoices read from file AMINTR to be sent to an EDI Trading Partner.

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.2.4.5

Coded name:

Name: BAT Generate Invoice - ECARS

Comment: @Definition: The batch process of generating invoice formats required for electronic billing in response to ECARS closing a rental. This process is triggered by TIME.

@Purpose: To be able to generate invoices for closed rentals in an automated fashion.

Process

Hierarchical numeric ID: 1.1.2.4.5.1

Coded name: RB0010

Name: PGM Create ECARS Invoice Detail (RB0010)

Comment: @Purpose:

To generate the initial billing invoices for closed rental contracts/tickets that have billing customers assigned that have a balance due that were closed on the previous business day.

Will also generate duplicate billing invoices for any closed rental contracts/tickets that have billing customers assigned that have a balance due that were selected for rebilling.

@Operational Method:

- IF on "DEV" development platform, then retrieve the associated host platform Machine Name for the rental contract/ticket's Group and Branch ID.

ELSE retrieve the current application host platform's Machine Name where this program is executing.

- Set any special machine related attributes.

- For each customer's closed rental tickets selected for generation of the billing invoices in the job file, RACBILLJ (ECR: Facsimile Billing Master File, sorted by Customer), perform the following steps:

1) Retrieve the closed rental contract/ticket information from the closed closed rental contract/ticket and its supplementary files.

2) Retrieve the ARMS profile and attempt to retrieve an associated ARMS Rental Transaction Cross-Reference information by executing the Retrieve ARMS Data (AM2090V1) program.

3) Check if the Bill-To Customer attributes returned from AM2090V1 indicates that the trading partner has the capability to receive electronic billing invoices or should be sent paper billing invoices. (Paper bills are default.)

4) IF electronic billing for ARMS,

a) Generate formats APPD01, IEBH01, ELCD01, RNTD01, REND01.

b) IF the profile dictates that the RATD01 format needs to be generated, generate it with 3 different information contents - units and rates, units only and units and rates without relating to each other.

c) Write the data queue entry to DQAM62V1 to send of the invoice transaction.

d) Delete the original bill from RACBILL.

e) Maintain electronic bills generation counts.

ELSE electronic billing for non-ARMS customer.

a) Generate formats APPD01, IEBH01, ELCD01, RNTD01, and REND01.

b) For invoicing only customers also generate the AUTD01, and ADJD01 formats.

ARMS Process Report

c) Write the data queue entry to DQAM6 to send the invoice transaction.

- d) Delete the original bill from RACBILL.
- e) Maintain electronic bills generation counts.

ELSE paper bills

a) Print the bill to appropriate medium (tape file, physical file)
(Tape is turned over to a third party firm for sorting by ZIP/Postal Code, grouping and printing, envelope stuffing, postage and mailing.)

- b) Delete the original bill from RACBILL if bill was printed.

- c) Maintain paper bill generation counts

5) If unsupported version of billing type (type greater than 4), print on exception report and update the bill in RACBILL to a daily frequency so that the bill may be generated the day the error has been fixed.

6) IF ticket is not found in the closed ticket files, print on exception report and delete the bill from RACBILL.

7) At customer total time, maintain number of envelopes needed and reset paper billing counter.

- Repeat the above steps for the next customer until there are no more customers to process.

@Notes:

The various ticket types are : I=Insurance B=Bodyshop D=Dealer R=Regular

0=Other.

The various billing types are : 1=Tot Less 2=Dy Tax 3=Dy NoTax 4=Other

5=Rebill.

The ARMS system now enables electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer). Therefore, at the initial billing invoice generation occurrence, no ARMS Rental Transaction Cross-Reference file record will exist at that time.

@Files: (CRUD)

RACBILLJ	(R)
RABMSG	(R)
FEDTAXID	(R)
QRACCLSQ	(R)
RACSCLD	(R)
OFFDRB	(R)
CUSTMAST	(R)
ECSURCHG	(R)
RACBRIST	(R)
RACBILL	(RU)
RACINS	(R)
BCPECG	(R)
AMXCLS	(R)
OFGBLN	(R)
RACILN	(R)
RACJD	(R)
RACBILLT	(C)
AMINTR	(CRU)
RAGBBT	(R)

Process

Hierarchical numeric ID: 1.1.2.4.5.2

Coded name: CHKZP

Name: PGM Format ZIP (CHKZP)

Comment: @Purpose: To place the passed Zip Code into proper country format for printing or display.

@Operational Method:

This program expects the following parameters: (EDS CHKZPI)

ZPZIPS - SHIPPING Zip Code

ZPEDTS - Edit Code Type

If not provided, format of edited zip code is determined by state.

If provided, determines the format desired for the passed zip code and ignores state code. Valid action codes are:

1:	5 dash 4	(xxxxx-xxxx)	U.S.
2:	5 blank 4	(xxxxx xxxx)	U.S.
3:	** NOT USED **		
4:	5 only	(xxxxx)	MEXICO
5:	3 blank 3	(xxx xxx)	CANADA
6:	4 blank 3	(xxxx xxx)	EUROPE
9:	9 blank	(xxxxxxxxx)	GENERIC

ZPSTS - State. If passed, and edit code is blank, is used to retrieve proper edit format from state code file.

If state not found, edit format defaults to #9.

ZPNUMS - Edit Code (Blank, "A", "B", "N" or "Z")

Blank = A thru Z & 0 thru 9

A = A thru Z

B = A thru Z, 0 thru 9 & blank

N = 0 thru 9

Z = Anything

ZPZIPM - MAILING Zip Code

ZPEDTM - Edit Code Type

If not provided, format of edited zip code is determined by state.

If provided, determines the format desired for the passed zip code and ignores state code. Valid action codes are:

1:	5 dash 4	(xxxxx-xxxx)	U.S.
2:	5 blank 4	(xxxxx xxxx)	U.S.
3:	** NOT USED **		
4:	5 only	(xxxxx)	MEXICO
5:	3 blank 3	(xxx xxx)	CANADA
6:	4 blank 3	(xxxx xxx)	EUROPE
9:	9 blank	(xxxxxxxxx)	GENERIC

ZPSTM - State. If passed, and edit code is blank, is used to retrieve proper edit format from state code file.

If state not found, edit format defaults to #9.

ZPNUMM - Edit Code (Blank, "A", "B", "N" or "Z")

Blank = A thru Z & 0 thru 9

A = A thru Z

B = A thru Z, 0 thru 9 & blank

N = 0 thru 9

Z = Anything

ARMS Process Report

The program returns the following parameters: (EDS CHKZPO)

ZOZIPS - SHIPPING Zip code, formatted for print. 10 pos.
ZOCNTS - Country code. Retrieved from State file if not passed.
ZONAMS - Country Name. Retrieved from State file.
ZOMSGS - Error Message. If field is greater than blank, indicates an error.
ZOZIPM - SHIPPING Zip code, formatted for print. 10 pos.
ZOCNTM - Country code. Retrieved from State file if not passed.
ZONAMM - Country Name. Retrieved from State file.
ZOMSGM - Error Message. If field is greater than blank, indicates an error.

@Notes: Program moves input ZIP to output ZIP before performing edits.

This serves as a default in case that an error occurs, program will always return a value. Calling program should but is not forced to look at the error message.

Process

Hierarchical numeric ID: 1.1.2.4.5.3

Coded name: VTS012A

Name: PGM Retrieve Vehicle Licensing Fee Charge Detail (VTS012A)

Comment: @Purpose:

To retrieve the Vehicle License Fee Days, Daily Unit Charge, and Total Charge for all input units for a group, branch, and ticket number.

@Operational Method:

The program expects the following data elements as the parameters (EDS VT024X1) with group, branch, ticket and the four unit numbers filled in:

Processing Mode
Ticket
Calendar Rental Flag
Ticket End Date
Renting Group
Renting Branch
Init Total Days
Unit Total Days
Unit 1
Unit 2
Unit 3
Unit 4
Date Out 1
Date Out 2
Date Out 3
Date Out 4
Date In 1
Date In 2
Date In 3
Date In 4
Init VLF Days Unit 1
Init VLF Days Unit 2
Init VLF Days Unit 3
Init VLF Days Unit 4
New VLF Days Unit 1
New VLF Days Unit 2
New VLF Days Unit 3

New VLF Day Unit 4
 Segment 1 Days
 Segment 2 Days
 Segment 3 Days
 Segment 4 Days
 Segment 1 Weeks
 Segment 2 Weeks
 Segment 3 Weeks
 Segment 4 Weeks
 Segment 1 Months
 Segment 2 Months
 Segment 3 Months
 Segment 4 Months
 Daily VLF Unit 1
 Daily VLF Unit 2
 Daily VLF Unit 3
 Daily VLF Unit 4
 Total VLF Unit 1
 Total VLF Unit 2
 Total VLF Unit 3
 Total VLF Unit 4
 Total Rental VLF
 CA Branch
 Employee Number
 Start Charges 1 Time
 Ticket end time
 VLF Bill To Flag
 CA License Renewal No-Unit 1
 CA License Renewal No-Unit 2
 CA License Renewal No-Unit 3
 CA License Renewal No-Unit 4
 Bill To Less VLF
 Bill To Plus VLF
 Return Code 1
 Return Code 2
 Return Code 3
 Return Code 4

Process

Hierarchical numeric ID: 1.1.2.4.5.6

Coded name: CCRAVN

Name: PGM Retrieve VIN (CCRAVN)

Comment: @Purpose: To return the Vehicle Identification Number (VIN) for a unit.

@Operational Method:

The program expects the following input parameters: unit number, ticket type ('D'ealer) and Option ('1'= Open a Ticket). VIN is returned along with the print flag. The parameter list is as follows:

UNIT NUMBER
 TICKET TYPE
 VIN#
 PRINT FLAG
 TICKET OPTION

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.2.4.5.7

Coded name:

Name: RPT Exceptions of ECARS Rental Contracts Not Invoiced

Comment: @Purpose: To create a report of all the ECARS closed tickets that were not invoiced during this invocation of the invoice generator, RB0010, because of an error so that it can be investigated and the error fixed before the ticket is resubmitted for invoicing.

@Report Distribution: Not defined.

Process

Hierarchical numeric ID: 1.1.2.4.5.8

Coded name: CLJ340

Name: PGM Driver Program for Invoice Generation (CLJ340)

Comment: @Purpose: Prepare and bill closed / re-billed tickets for all trading partner based on their frequency.

@Operational methods:

- Collect requests for any ticket to be re-billed.
- Total the number of records for a given control group
- Update the detail records with the total for the control group
- Generate the invoices
- Generate listing of invoices not created.

Process

Hierarchical numeric ID: 1.1.2.4.5.9

Coded name: RB0030

Name: PGM Create Total Record for Control Group (RB0030)

Comment: @Purpose: Create a total record for each control group, to be used later in the process for sorting and processing.

@Operational Method:

- Read every detail record, incrementing a number per control group.
- Write a record to work file with the total for control group.

@Notes: This pertains to invoices that are faxed.

Process

Hierarchical numeric ID: 1.1.2.4.5.10

Coded name: RB0031

Name: PGM Update Detail Records with Control Group Total (RB0031)

Comment: @Purpose: Update detail records with control group total.

@Operational Method:

- Read every detail record, updating it with the control group total

@Notes: This pertains to invoices that are faxed.

ARMS Process Report

Process

Hierarchical numeric ID: 1.1.2.4.5.11

Coded name: RB0037

Name: PGM Calculate the actual number of Facsimilie (RB0037)

Comment: @Purpose: Update the individual detail records with the number total number control records for a givien group.

@Opertaional Method:

- Read the file counting the number of control records for a given group
- When the control group changes, update each detail record with that number.

@Notes:

- This number is used for determining if a cover page is to be generated with the invoices.

Process

Hierarchical numeric ID: 1.1.2.4.5.12

Coded name: F20114

Name: PGM Generate Facsimile Spool File for Control Group (F20114)

Comment: @Purpose: Distribute the spool file based on language code

@Operational Method:

- Read to work file group all records for a customer together.
- When the total page count is less than 6 no cover page is generated.
- When the total page count is 6 - 9, generate cover page - Portrait.
- When the total page count is greater than 10, generate cover page - landscape.
- Generate report.

Process

Hierarchical numeric ID: 1.1.2.5

Coded name:

Name: BT Close ARMS Authorized Ticket

Comment: @Definition: The set of online activities that are used to calculate the billing totals for an ARMS Authorized Ticket that has closed.

@Purpose: Process that is executed to calculate the Trading Partner amount due.

Process

Hierarchical numeric ID: 1.1.2.5.2

Coded name:

Name: ONL Close ARMS Authorized ECARS Ticket

Comment: @Definition: The process that close goes through to calculate the billing charges for an ARMS ticket.

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.2.5.2.1

Coded name: ECARMSV1

Name: PGM Calculate EC Billing Charges for ARMS Tickets (ECARMSV1)

Comment: @Purpose: To calculate the amount to be billed to the renter and the amount to be billed to the insurance company at closing time for an ARMS authorized ticket.

@Operational Method:

This program when called will receive the ticket information as parameters.

For the passed ticket, perform the following steps:

- 1) Calculate the number of days to bill and total days authorized.
- 2) Get the group branch tax information to determine what surcharges apply and if the surcharges are taxable.
- 3) Load the approved surcharges from the file ECSURCHG, Surcharges Approved by Insurance Company File, into arrays based on the surcharge type, i.e., daily, lump and percent. Add the daily rate augmentor surcharges ('02'=city/state and '20'= underage) to the calculated authorized daily rate. Similarly, add the lump surcharge with code '11'=Miscellaneous to the calculated authorized lump total field. These special surcharges are not loaded to the surcharge arrays.
- 4) Check for any government surcharge, airport charge, additional charges (baby seat, additional driver charge etc) per rental charged on the ticket against the lump array to see if the insurance company has authorized that surcharge. If the amount in the ticket information matches the amount in the array, it means that this surcharge amount was authorized by the insurance company as the arrays were loaded from the approved surcharges file. If authorized, add this to the calculated lump total or the taxable lump total field if the charge is taxable for this ticket.
- 5) Check for any government surcharge, airport charge, additional charges (baby seat, additional driver charge etc), damage waiver for each unit, personal accident insurance for each unit, per day, charged on the ticket against the daily array to see if the insurance company has authorized that surcharge. If the amount in the ticket information matches the amount in the array, it means that this surcharge amount was authorized by the insurance company as the arrays were loaded from the approved surcharges file. If authorized, add this to the calculated daily authorized rate total or the dialy taxable total field if the charge is taxable for this ticket.
- 6) Similarly, check for percentage surcharges in the percent array.
- 7) Apply any discount to the calculated rates and compare against authorized maximum amounts for the ticket. If the calculated charges exceed the maximums, select the maximums as the charges to be billed to the insurance company.
- 7) If the claim type is "claimant", policy maximums are ignored otherwise the lesser of the bill-to amount and the policy maximum is chosen.
- 8) Retrun with the amount to be billed to the insurance company and the amount to be billed to the renter.

@Notes: Some ARMS Trading Partner Companies does not send us the amounts for the surcharges. They simply send us flags indicating which surcharges they will pay for ,e.g., PAI, DW, SLP etc. Any charge that the branch charges for the surcharge flag sent by them will be considered authorized subject to the maximums.

@Files:

GPBRTX	(_R_)
ECSURCHG	(_R_)
VT010P00	(_R_)
OFFDRB	(_R_)

Process

Hierarchical numeric ID: 1.1.2.5.3

Coded name:

Name: ONL Close Claims Connection Ticket

Comment: @Definition: The process that close goes through to calculate the billing charges for an ARMS ticket.

Process

Hierarchical numeric ID: 1.1.2.5.3.1

Coded name: CCARMSV1

Name: PGM Calculate CC Billing Charges for ARMS Tickets (CCARMSV1)

Comment: @Purpose: To calculate the amount to be billed to the renter and the amount to be billed to the insurance company at closing time for an ARMS authorized ticket.

@Operational Method:

1) Retrieve the ticket data.

2) Determine how many and which rental locations were used.

3) For each location perform the following steps:

- * Calculate number of billing days for this location.
- * Determine Bill-To days and the remaining authorized days
- * Determine Bill-To Daily Rate
- * Calculate total surcharge amount for this ticket/location
- * Calculate Daily Maximum
- * Select the lesser of the daily rate or the daily maximum
- * Multiply bill-to daily rate times the number of bill-to days to determine extended bill-to rate.
- * Add taxable authorized lump totals
- * Calculate tax percentage. Multiply extended bill-to rate time tax percent to determine bill-to taxable amount
- * Calculate markup per day amount, extend it and add to the bill-to amount

- * Add in percentage surcharges to total bill-to.

- * Percentage surcharges are non-taxable and are applied only to the base amount

* Add approved non-taxable lump, daily, and percentage surcharges to the total bill-to ticket amount.

* If loss type is NOT a claimant, select the lesser of the authorized amount and the calculated bill-to amount.

* Calculate the bill-to percentage and the amount due from the trading partner.

* If loss type is NOT a claimant, select the lesser of the amount due from the trading partner or the policy maximum.

@Files:

CCCLSC (_R_)
CCMASTER (_R_)
CCSURDTL (_R_)

ARMS Process Report

Process

Hierarchical numeric ID: 1.1.3

Coded name:

Name: AA Office Information Synchronization Between Enterprise and Trading Partner
Comment: @Definition: The application area responsible for keeping the Enterprise office information database on the insurance companies host computer in synch with the database on Enterprise computer with regard to any new additions, changes or deletions to the office or the predefined rental rates information.

Process

Hierarchical numeric ID: 1.1.3.1

Coded name:

Name: BT Synchronize Office Information for specified ARMS Trading Partner Companies
Comment: @Definition: The set of automated activities that are used to create and transmit Rental Location office information to a specified ARMS Trading Partner Company's host computer.

@Purpose: Communicate Enterprise existing rental location office information (telephone and/or rates) to the trading partner.

Process

Hierarchical numeric ID: 1.1.3.1.5

Coded name:

Name: BAT Generate Office Maintenance Transactions for specified ARMS Trading Partner Companies

Comment: @Definition: The batch process that generates required office and predefined rental rates maintenance transactions to be sent to specified ARMS Trading Partner Companies host computer. This batch is triggered by TIME and scheduled in the automated job scheduler.

@Purpose: To allow for timely updated information of office and rate info to specified ARMS Trading Partner Companies.

@Notes: Rental Location Office changes are made by the Rental Application Systems during the day. These changes are process when this processes runs.

Process

Hierarchical numeric ID: 1.1.3.1.5.1

Coded name: CLL900

Name: PGM Distribute Office and Rate Updates (CLL900)

Comment: @Purpose:

To collect office and predefined rental rates information changes and generate maintenance transactions to synchronize this information on the insurance company's host computer. Also, to consolidate these transactions from the distributed systems to the centralized ARMS system.

@Operational Method:

1) CALL CLL042 to generate rate maintenance transactions in file AMLOCTRN, Location/Rate Change Transactions.

ARMS Process Report

2) IF the returned run status is 'Y', CALL AM041V1 to generate the ARMS proprietary formats and send them to the centralized system.
ELSE end the program.

Process

Hierarchical numeric ID: 1.1.3.1.5.2
Coded name: CLL042
Name: PGM Gather Rate Updates (CLL042)
Comment: @Purpose:

To determine if there are any changes to the predefined rental rates in file NRXRATES (national reservation exception rates) for insurance type customers and to generate synchronizing transactions for maintaining the rates on specified ARMS Trading Partner Company's host computer.

@Operational Method:

- Retrieve job run date and add one day to get tomorrow's date. Subtract one day to get yesterday's date.

- 1) Retrieve the last job completed date from data area DAL900STS and compare this date to yesterday's date.
 - a) IF this is not a rerun and the dates are not equal, notify ARMS On-Call (backup and primary). Set the run status flag to 'N' and return, so that the calling program, CLL900, does no further processing.
 - b) IF this is a rerun or the dates are equal (i.e. the job was completed yesterday), continue.
- 2) Open a query file selecting those rates from NRXRATES that become ineffective today and those that become effective tomorrow. Then CALL AM0042 to create rate maintenance transactions in AMLOCTRN for these selected records.

Process

Hierarchical numeric ID: 1.1.3.1.5.3
Coded name: AM0042
Name: PGM Create Rate Update Transactions (AM0042)
Comment: @Purpose:

To create predefined rental rate maintenance transactions for a location. Each location may have rates specified for each type of car that the location hosts with effective dates.

@Operational Method:

- 1) Read NRXRATES (Query file over NRXRATES, national reservations exception rates opened by CLL042) in a loop until end of file. The selection criteria for the query are: {Pre-arranged discount type = 'I' (Insurance)} AND {(Start Date = Job Date + 1) OR (Stop Date = Job Date)}. This will select any rates for a location that are ending today and/or any rates that are effective tomorrow.
- 2) In order to determine which type of maintenance (add, change or delete) to perform; this program holds the previous records key while it sequentially reads the query file (Key: Group + Branch + Car Type + Start Date). Remember we are dealing with only those records that are selected by the query.

ARMS Process Report

a) IF the current record's Group+Branch+Car Type matches the previous record's, then this signifies a 'C'hange. Create rate change transaction in AMLOCTRN for this location and car type.

b) IF the current record's Group+Branch+Car Type does not match the previous record's, then :

i) IF the start date is tomorrow, this is an 'A'dd. Create rate add transaction in AMLOCTRN for this location and car type.

ii) IF the stop date is today, this is a 'D'lete. At this point, this condition HAS to be true. Create rate delete transaction in AMLOCTRN for this location and car type.

@Files:

NRXRATES (_R_)
NRCTY (_R_)
AMLOCTRN (C_)

Process

Hierarchical numeric ID: 1.1.3.1.5.4

Coded name: AM0041V1

Name: PGM Format and Distribute Office and Rate Updates (AM0041V1)

Comment: @Purpose:

To read the office and rate change transactions and generate appropriate ARMS proprietary formats for the synchronizing maintenance transactions that need to be sent to the insurance customer.

@Operational Method:

1) Read the ARMS Profile Application Specific Data (ARMSPR1) file starting with the first record and for each profile that specifies that the insurance company receives office and rate updates, perform the following steps.

2) Read the AMLOCTRN file and for each location in this file, check to make sure that the location is set up to receive ARMS reservations. IF the location is not set up, skip this record and read the next record ELSE continue.

3) Get the effective predefined rental rates for this location from the NRXRATES file and generate the office and rate maintenance transaction sets (format OFFD01) for this profile and write to file ANDIST for transmission to the centralized system. A data queue entry to the data queue DQANDST will trigger the transmission.

4) Get next record from AMLOCTRN and repeat steps 2,3,4 until all records in AMLOCTRN are read.

5) Finally, delete all records from file AMLOCTRN.

@Files:

ARMSPR1 (_R_)
AMLOCTRN (CR_D)
NRXRATES (_R_)
DROFF (_R_)
NRCTY (_R_)
ANDIST (C_)

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.3.2

Coded name:

Name: BT Synchronize Office Information for X12 Customer

Comment: @Definition: The set of automated activities that are used in the creation and transmission of Rental Location Office information to be sent to X.12 customers .

@Purpose: Communicate Enterprise existing rental location office information to the trading partner.

Process

Hierarchical numeric ID: 1.1.3.2.6

Coded name:

Name: BAT Generate Office Location / Rates Maintenance Transaction Set 832 for X12

Comment: @Definition: The batch process of generating required office maintenance transactions (which includes office information and rates information) to be sent to X.12 customers. This batch is triggered by TIME and scheduled in ROBOT.

@Purpose: To allow for timely updated information of office and rate info to X-12 customers.

Process

Hierarchical numeric ID: 1.1.3.2.6.1

Coded name: AML832A001

Name: MOD Convert Journal Entries to Transactions (AML832A001)

Comment: @Purpose:

To convert office directory information journaled changes to a change transactions for further processing.

@Operational Method:

- Read the job file, AML832JA (created via the DSPJRN command in the calling procedure) sequentially.

- Based on the before and after images of the office directory records in the journal entries, construct the change transactions with the appropriate business case into file, AM830P (AM Office change transactions). This will handle two business cases :

- 1 = Add/Update Rental Car Company Branch Number
- 6 = Delete Branch Number

@Notes:

1. Only U.S. Daily Rental locations will be processed.
2. If an error occurs, generate an error/page, then return and 'Exit Code' of '1' so the caller can detect that an error occurred.

Note also that this program module reads from AML832JA which is a file created by the DSPJRN command.

@Files:

AM830P (C_UD)
MACHID (R_)

ARMS Process Report

Process

Hierarchical numeric ID: 1.1.3.2.6.2
Coded name: AML832A002
Name: MOD Determine Company & Invoke Send Office Updates (AML832A002)
Comment: @Purpose:

To determine which companies have requested for the office and pre-defined rental rate change transactions to be sent to them and to send the change transactions to the company if requested.

@Operational Method:

1) Read the profile data to determine which company has requested to receive the X.12 synchronizing office and rate maintenance transactions.

2) For each company that has requested office updates, perform the following (procedure PrcCstOfcUpdts):

a) Build the beginning transmission envelope using procedures BldArmsKey, BldTSMS01, BldGRPS01 and BldSETS01 in that order into AMSEND.

b) Invoke the GetCarClasses procedure with appropriate parameters to get the Enterprise and X12 car classes for this customer from file AMCLSTBL.

c) For all records in file AM830P (Office change transactions), invoke the procedure FmtSndOfcUpdts with appropriate parameters to write office update records to AMSEND. If a new office location is being added and the GetCarClasses procedure returned car classes, invoke the FmtSndRateUpdts procedure with appropriate parameters to write Rate Format records to AMSEND for the customer.

d) Build the ending transmission envelope using procedures BldSETE01, BldGRPE01 and BldTSME01 in that order into AMSEND.

e) Send a Data Queue to DQ832S to initiate the sending of this transmission.

@Notes:

If an error occurs, generate an error/page, then return an 'Exit Code' of '1' so the caller can detect that an error occurred. If an 'Exit Code' of '1' is returned from a called procedure, return the 'Exit Code' to the caller immediately.

@Files:

AM830P (-R--)
AM831P (-RU-)
ARMSPR1 (-R--)
ARMSPR3 (-R--)
AMSEND (C--)

Process

Hierarchical numeric ID: 1.1.3.2.6.3
Coded name: AML832A003
Name: MOD Format and Send Office Updates (AML832A003)
Comment: @Purpose:

ARMS Process Report

To format the information passed from the transaction file record to prepare it for sending.

@Operational Method:

Procedure FmtSndOfcUpdts

1) Use procedure BldArmsKey to build the key for AMSEND.

2) Use procedure BldLocFmt (Build location format) to build the office maintenance transaction proprietary format X832F2 and write to AMSEND.

3) IF the location format is a business case 6 (delete branch), additional formats need to be generated to handle the service area updates as a result of the delete.

a) Retrieve all the service areas served by this branch (rental location) by reading the service area/location file AM831P01 in a loop using the rental location as the key.

b) For each of these service areas, call the locator procedure, AMRtnUSLoc, passing the service area to get the new branch for this service area. Update service area file, AM831P01 with the new location returned from AMRtnUSLoc unless a return code of '06' is received or the location returned is blank. In these cases, generate an error/page and skip to the next service area.

c) Use procedure BldSvcFmt to create the service area update format X832F1 and write to AMSEND.

@Files:

AMSEND (C---)
AM831P (-RU-)

@Notes:

1. The service area formatting is shared with job AML833A and as such must be considered for a service program.

2. In general, if an error occurs generate an error/page; then return an 'Exit Code' of '1' so the caller can detect that the error occurred. If an 'Exit Code' of '1' is returned from a called procedure, return the 'Exit Code' to the caller immediately. There may be alternative handling of some errors noted in the operational method.

Process

Hierarchical numeric ID: 1.1.3.2.6.4
Coded name: AML832A004
Name: MOD Maintain data area AM832A (AML832A004)
Comment: @Purpose:

To maintain data area AM832A with the starting (from) date and time to be used for filtering the DSPJRN command in program AML832A on the next scheduled run. This is the last task performed after all processing for the job has been completed.

@Operational Method:

Procedure SetNxtFrmDtTm.

Retrieve and lock data area AM832A. It will contain the starting (from) date and time used in the DSPJRN command for the current job.

Add 1 second to the date and time passed to this procedure, which

ARMS Process Report

will contain the ending (to) date and time used to filter the DSPJRN command for the current job.

Update data area AM832A with the new value and unlock.

@Notes:

In general, if an error occurs generate an error/page; then return an 'Exit Code' of '1' so the caller can detect that the error occurred.

Process

Hierarchical numeric ID: 1.1.3.2.6.5
Coded name: AML832A005
Name: MOD Format and Send Rates/Car Classes (AML832A005)
Comment: @Purpose:

To format the rates for retrieved car classes for any new Enterprise location being sent to the specified insurance company.

@Operational Method:

Procedure FmtSndRateUpdts.

1) Use procedure GetClassRates to get the rates for the specified customer for the specified location.

2) Use procedure SndRateFmts to write rate format X832F3 records to AMSEND for each X12 Class and Rate. Handle return codes as follows:
ALLRATES '00' Rates for all classes were found and sent for loc.
INVALPARM '01' Generate error/page and process next location.
NORATES '02' Generate error/page and process next location.
SOMERATES '03' Rates for some of the classes were found and sent.
LOCKEDRECORD '99' Generate error/page and process next location.

@Files:

AMCLSTBL (-R--)
AMSEND (C---)

@Notes:

1) In general, if an error occurs generate an error/page; then return an 'Exit Code' of '1' so the caller can detect that the error occurred. There may be alternative handling of some errors noted in the operational method.

Process

Hierarchical numeric ID: 1.1.3.2.6.6
Coded name: AML832A006
Name: MOD Build ARMS Start/End for Office Updates (AML832A006)
Comment: @Purpose:

ARMS Process Report

To build the key and data fields for the transmission, group, and set start and end envelope for office and rate updates.

@Operational Method:

The following procedures are exported for use by other procedures. When building the office and rate update transactions, these procedures provide the ARMS proprietary transmission envelope formats.

The file AMSEND consists of two parts, the key and the proprietary format. These procedures will return the key and envelope proprietary formats in the AMSEND data structure, AMSENDDS.

1) Procedure BldArmsKey:

- a) Clear the ARMSKEY and AMSEND data structures.
- b) Populate the ARMSKEY (Data Structure) fields and move the ARMSKEY data structure to the ASNKEY field of the AMSEND data structure
- c) Return

2) Transmission Start procedure BldTSMS01:

- a) Clear the TSMS01 data structure.
- b) Populate the TSMS01V1 (Data Structure) fields and move the TSMS01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

3) Group start procedure BldGRPS01:

- a) Clear the GRPS01 data structure
- b) Populate the GRPS01V1 fields and move the GRPS01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

4) Set start procedure BldSETS01:

- a) Clear the SETS01 data structure
- b) Populate the SETS01V1 fields and move the SETS01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

5) Transmission end procedure BldTSME01:

- a) Clear the TSME01 data structure
- b) Populate the TSME01V1 fields and move the TSME01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

6) Group end procedure BldGRPE01:

- a) Clear the GRPE01 data structure
- b) Populate the GRPE01V1 fields and move the GRPE01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

7) Set end procedure BldSETE01:

- a) Clear the SETE01 data structure
- b) Populate the SETE01V1 fields and move the SETE01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

@Notes :

1. This module has been bound by copy into two different ILE programs AML832A and AML833A and such must be converted into a service program at some point.

ARMS Process Report

2. If an error occurs, generate an error/page, then return an 'Exit Code' of '1' so the caller can detect that an error occurred.

Process

Hierarchical numeric ID: 1.1.3.2.6.7
Coded name: AML833A001
Name: MOD Select Unique Service Areas (AML833A001)
Comment: @Purpose:

To select each unique ARMS Trading Partner Company service area (area code + exchange) from all the phone numbers in the US, associate a location, and record the combination.

@Operational Method:

1) The first parameter to this module specifies the service area to start processing from. If it is blank, process all service areas.

2) Since the service areas consist of the phone number, area code, and prefix; the maximum service area is '999999'. For each sequential possible service area from 1 through 999999, try positioning file AT001P01 to the record for that service area. If successful, send a data queue containing that service area to DQ834Q.

3) Monitor data queue DQ834Q while processing. Any time the data queue accumulates more than 1000 entries, delay the job for 45 seconds repeatedly until there are 1000 or less entries. If 30 delays are executed before the count falls to 1000 or less, return a value of 'Y' (Abnormal end) to AML833A so it will generate an error/page.

4) After service area '999999' is processed, return to AML833A.

@Files:

AT001P01 (-R--)

Process

Hierarchical numeric ID: 1.1.3.2.6.9
Coded name: AML833A003
Name: MOD Determine Company and Invoke Send SA Updates (AML833A003)
Comment: @Purpose:

To determine which companies have requested to receive the service area updates and then invoke the send routine for each such company.

@Operational Method:

1) Determine which company has requested to receive office and rate changes and for each such company, invoke procedure SndSvaUpd to send the updates.

@Notes:

If an 'Exit Code' of '1' is returned from a called procedure or an error occurs, generate an error/page and end the program immediately.

@Files:

ARMS Process Report

ARMSPR1 (-R--)
ARMSPR3 (-R--)

Process

Hierarchical numeric ID: 1.1.3.2.6.10
Coded name: AML833A004
Name: MOD Format and Send SA Updates (AML833A004)
Comment: @Purpose:

To format and send the service area change transactions to the specified company in the input parameters.

@Operational Method:

Procedure SndSvaUpd using specified company.

1) Use procedure BldArmsKey to build the key portion in file AMSEND.

2) Use transmission envelope procedures BldTSMS01, BldGRPS01 and BldSETS01 to build the beginning envelope.

3) For each service area in file, AM833P (AM Changed Service Areas) build proprietary format X832F1 with the appropriate business case into file AMSEND.

4) Build the ending envelope using procedures BldGRPE01, BldSETE01 and BldTSME01.

5) Send a data queue entry with the ARMSKEY to data queue DQ832S. This data queue entry will be picked up by the send interface program for actual mapping and sending.

6) Return to caller.

@Notes:

If an 'Exit Code' of '1' is returned from a called procedure or an error occurs, generate an error/page and end the program immediately.

@Files:

AM833P (-R--)
AMSEND (C---)

Process

Hierarchical numeric ID: 1.1.3.2.6.11
Coded name: AML833A005
Name: MOD Return a U.S. Location (AML833A005)
Comment: @Purpose:

To return a U.S. location for a passed service area (Area Code and Exchange)

@Operational Method:

1) Initialize the data structure that is passed to the locator program, RAS013A.

2) Execute appropriate overrides to make the locator retrieve Enterprise branches only.

3) Use the locator program to determine the best case rental branch that could serve the service area passed to the program.

5) To determine if the returned branch is usable, we check DROFLF1.
 IF prevent ARMS, use forward to location.
 IF No forward to location, keep looking.
 IF country code is 'CA', keep looking.

6) The return value for procedure AMRtnUSLoc should be 12 characters. The first 2 positions are the return code (same codes as RAS013A). The other 10 positions are the location (as returned from RSA013A).

7) If there is an unexpected error, use *PSSR to page on call staff after dumping the program

8) RETURN.

@Files:

DROFLF1 (-R--)
 DROFLF7 (-R--)

Process

Hierarchical numeric ID: 1.1.3.2.6.12

Coded name: AML833A

Name: PGM Send Service Area Updates (AML833A)

Comment: @Purpose:

To control the various procedures to synchronize the service areas (area code + exchange) representing Enterprise locations in the ARMS Trading Partner Company database with the Enterprise office directory database.

@Operational Method:

1) The first input parameter specifies the number of never ending jobs which this program submits; each of these jobs calls program AML834A (Process Service Areas). These jobs will share the processing of entries sent to data queue AM834Q by this program (module AML833A001) by repeatedly receiving the next available data queue entry and processing it. This method is used to speed up processing.

2) If the second parameter is blank, this is not a restart; clear files AM833P (changed service areas) and AM834P (Claims Connection service areas).

3) Once the jobs are submitted, call procedure AML833A001 to process service areas. The second parameter to this program specifies the starting service area from where this procedure will start processing in the global phone number file. If it is blank, all service areas will be processed. The procedure will sequentially select all service areas from the starting point and send them to data queue AM834Q for processing by the AM834A jobs.

4) Generate an error/page if AML833A001 returns a value of 'Y' (Abnormal end). Otherwise, send a *DOWN data queue to each of the AML8334A jobs to end them.

5) End this program when all the submitted jobs have shutdown.

@Notes:

- 1) This job runs only on the ARMS Centralized system.
- 2) This program clears file AML833DA but the file is not used for anything. It is in the evolution plan to remove it and update the comments in all associated modules.
- 3) If an error occurs or an 'Exit Code' of '1' is returned by a called procedure, generate an error/page and end the program immediately.

Process

Hierarchical numeric ID: 1.1.3.2.6.13
 Coded name: AML832A
 Name: PGM Update Office Directory (AML832A)
 Comment: @Purpose:

This is a CLLE program module and will serve as the entry point module for the ILE program AML832A which will consist of bound modules that will synchronize the office information between Enterprise and X.12 customers (i.e., any ARMS Trading Partner Company).

@Operational Method:

- 1) Make sure the machine that this module executes is "RARMS". If not RARMS, end the program.
- 2) Get the last process date and time for the job from the data area AM832A. This is where the last process ran left off and this run should now get the entries from this date and time to the current date and time.
- 3) Convert the journal entries available in the OMS journal, @JRNLIB/OMSJRN for file OFFDRB to a job file, AML832JA. The job file has been created using command DSPJRN and essentially has the format the same as QADSPJRN (IBM supplied outfile format) except that the last field called JOESD (Entry Specific Data) is of length 480 which is the record length for OFFDRB. The command syntax will be as follows:

```
DSPJRN JRN(@JRNLIB/OMSJRN) FILE((*LIBL/OFFDRB *FIRST)) RCVRNG(*CURCHAIN)
FROMTIME(&FROMDATE &FROMTIME) TOTIME(&TODATE &TOTIME) JRNCDE((R *ALLSLT))
OUTPUT(*OUTFILE) OUTFILFMT(*TYPE1) OUTFILE(QTEMP/AML832JA) ENTDTALEN(480)
```

The field JOESD will contain the after image of the entire OFFDRB record. Journal code (JRNCDE) of "R" means only those journal entries that relate to operations at the record level (add, update, delete). The journal receiver range needs to be *CURCHAIN which means that all the receivers that are ON LINE since the last break in the receiver chain. OMS deletes OMSJRN journal receivers when they are 48 hours old. A new journal receiver is created whenever the current receiver is 200 Megabytes in size and a new receiver is created every day whether the last receiver reaches its threshold or not.
- 4) CALLPRC CnvtJrnlToTrns (Module AML832A001) to convert the journal entries to change transactions.
- 5) CALLPRC PrcOfcUpdates (Module AML832A002) which will determine all the companies that have requested office information to be sent to them and will then invoke the send module for each company.

ARMS Process Report

6) CALLPRC S NextFrmDtTm (Module AML832A004) with the converted 'to date' and 'to time' in CYMD and HMS format. This procedure will maintain the data area AM832A for processing dates.

@Notes:

If an 'Exit Code' of '1' is returned from a called procedure or an error occurs, generate an error/page and end the program immediately.

This program module creates the file AML832JA as a result of the DSPJRN command.

Process

Hierarchical numeric ID: 1.1.3.2.6.15

Coded name:

Name: DTQ Service Areas to Send - Input to Sender (DQ832S)

Comment: @Definition: The data queue that initiates the sending of the X.12 synchronizing office and rate maintenance transactions to the X.12 insurance customer.

Process

Hierarchical numeric ID: 1.1.3.2.6.16

Coded name: AML834A

Name: PGM Process Service Areas (AML834A)

Comment: @Purpose:

To create a service areas update file containing service areas that have been changed since the last time these service areas were sent to the insurance customers.

@Operational Method:

- 1) Receive the service area from the input data queue, AM834Q.
- 2) Use procedure AMRtnUSLoc to get the Enterprise US location that will service this service area.
- 3) IF the service area exists in file AM831P (all US service areas file)
 - a) if the location on file is the same as the location returned by AMRtnUSLoc, skip this service area else update the file with the correct location.
 - b) add this service area to file, AM833P (changed service areas file)
 - c) if the returned location is Claims Connection, add the service area to file AM834P (Claims connection service areas).
- 4) IF the service area does not exist in file AM831P (all US service areas file)
 - a) add the service area to file AM831P.
 - b) add this service area to file, AM833P (changed service areas file)
 - c) if the returned location is Claims Connection, add the service area to file AM834P (Claims connection service areas).
- 5) When a shutdown entry is received from the data queue, update data area AM832A to reflect that this job is shutting down.

@Files:

AM831P (CRU-)
AM833P (CR--)
AM834P (C---)

Process

Hierarchical numeric ID: 1.1.3.2.6.17

Coded name:

Name: DTQ Service Areas to Process (AM834Q)

Comment: @Definition: This data queue is used to feed job AML834A with unique service areas for all US phone numbers. There can be several AML834A jobs running that feed off of this data queue.

Process

Hierarchical numeric ID: 1.1.3.3

Coded name:

Name: BT Send Initial or Yearly Car Class Rates and Initail Office Information for X12 Customer

Comment: @Definition

The initial/yearly process of sending car class rates and office information to insurance customers who recieve this information in X.12 format.

@Purpose

Provide insurance customers with Enterprise office information and car class rates initially. Provide them with car class rates both initially and at yearly intervals.

Process

Hierarchical numeric ID: 1.1.3.3.1

Coded name: AML831A

Name: PGM Send Car Class Rates (Initial/Yearly) and Office Info (Initial) (AML831A)

Comment: @Purpose

To control the procedure to send initial or yearly rates for car classes and initial office information to insurance customers who receive this information in X.12 format.

@Operational Method:

This program only runs on RARMS or DEV. On other computer platforms, it will exit immediately without processing.

Call procedure AML831A001 to retrieve rates and send them for the insurance customer specified.

Process

Hierarchical numeric ID: 1.1.3.3.2

ARMS Process Report

Coded name: AML831A001

Name: MOD Send Rates (Initial/Yearly) and Office Info (Initial) (AML831A001)

Comment: @Purpose

To send initial or yearly car class rates and initial office locations to the specified insurance customer in X.12 format.

@Operational Method

1) Retrieve a transmission ID from data area AMTCID.

2) Call procedures in module AML832A006 to build the key and write the start formats to file AMSEND.

3) Call procedure GetCarClasses (module AML832A005) to retrieve car classes for the customer.

4) Call procedure LoadMachArray to retrieve all U.S. computer platforms.

5) Read through file OFFDRB, and process only U.S. office locations.

6) Call procedure FmtSndRateUpdts (module AML832A005) to send rates for the car classes retrieved. These will be written to AMSEND in X832F3 formats. If this is the initial load, specify business case '1'. If this is a yearly load, specify business case '3'.

7) If this is the initial load, call subprocedure SndOfcUpdt to send the office location information. This will be written to AMSEND in X832F2 formats using business case '1'.

8) Call procedures in module AML832A006 to build the key and write the end formats to file AMSEND.

9) Send a data queue to DQ832S containing the key with the transmission ID to initiate sending the information to the insurance customer when batch program AMZ832A runs.

@Files: (CRUD)

OFFDBB (-R--)

ARMSPR1 (-R--)

ARMSPR3 (-R--)

MACHID (-R--)

AMSEND (C--)

Process

Hierarchical numeric ID: 1.1.4

Coded name:

Name: AA ARMS Support

Comment: @Definition: The application area responsible for supporting ARMS communications environment, producing reports for customers and handling any operational problems that are discovered.

Process

Hierarchical numeric ID: 1.1.4.1

ARMS Process Report

Coded name:

Name: SA Manage Environment

Comment: @Definition: The activity that initiates the ARMS application and performs a controlled shutdown when needed. It also involves tasks to handle transactions bound for a machine that is currently unavailable.

Process

Hierarchical numeric ID: 1.1.4.1.1

Coded name:

Name: BAT Purge Application Database

Comment: @Definition: This activity archives and reorganizes the application database, including some of the transactions work files.

Process

Hierarchical numeric ID: 1.1.4.1.1.1

Coded name: AML992

Name: PGM Purge Cross-Reference and Associated Database File(s) by Vendor Id (AML992)

Comment: @Purpose:

To maintain the ARMS database by purging (deleting) old records.

@Operational Method:

- Calculate the oldest date for retained data.
- For each cross-reference record:
 - Delete all records, in the database, associated to the cross-reference where the status code and date criteria are met.
- Delete records as above until a shutdown request is detected.

@Notes: files AM095P and AM096P are retained for 25 months.

@Files: CRUD

AMXREF	(-R-D)
AMMNTLOG	(-R-D)
AMAUTD	(-R-D)
AMADJD	(-R-D)
AMCOMD	(-R-D)
AMINS	(-R-D)
AMIEBH	(-R-D)
AMIEBD	(-R-D)
AMIEBT	(-R-D)
AMRATD	(-R-D)
AMRMTD	(-R-D)
AMRNTD	(-R-D)
AMRPRD	(-R-D)
AMSURD	(-R-D)
AMTIME	(-R-D)
AMMSCLOG	(-R-D)
AM990P	(-RUD)
AM095P	(-R-D)
AM096P	(-R-D)

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.4.1.1.3

Coded name: AML994A

Name: PGM Stop Cross-Reference and Associated Database File(s) Purge (AML994A)

Comment: @Purpose:

To change the control data area (AM006A) for the purge (program AML992A), instructing it to shutdown.

@Operational Method:

- Change the contents (position 1 - 8) of the data area to cause the program to shutdown.

Process

Hierarchical numeric ID: 1.1.4.1.1.4

Coded name: AML993A

Name: PGM Purge Orphan Transaction / Database Records (AML993A)

Comment: @Purpose:

To delete every record in the ARMS database, that does not have a corresponding cross-reference record.

@Operational Method:

- For each database file:
 - Read the file, deleting any record that does not have an associated cross-reference record.

@Note:

- Files AM0060SV, AMXFRLOG, and AMPGMERR are purged using a date obtained from the control file. Files AM095P and AM096P are retained for 25 months.

@Files

- AMERRHST	(-R-D)
- AM990P	(-R--)
AM0060SV	(-R-D)
AMADJD	(-R-D)
AMAUTD	(-R-D)
AMBILERR	(-R-D)
AMBILLOG	(-R-D)
AMCOMD	(-R-D)
AMIEBD	(-R-D)
AMIEBH	(-R-D)
AMIEBT	(-R-D)
AMINS	(-R-D)
AMMNTLOG	(-R-D)
AMMSCLOG	(-R-D)
AMPGMERR	(-R-D)
AMRATD	(-R-D)
AMRCVERR	(-R-D)
AMRMTD	(-R-D)
AMRNTD	(-R-D)
AMRPRD	(-R-D)
AMSNDLOG	(-R-D)
AMSTX	(-R-D)
AMSURD	(-R-D)
AMTIME	(-R-D)

ARMS Process Report

AMTRNLOG (-R-D)
AMXFRLOG (-R-D)
AMXREF (-R-D)
AM095P (-R-D)
AM096P (-R-D)

Process

Hierarchical numeric ID: 1.1.4.1.1.5

Coded name: AML995A

Name: PGM Stop Orphan Transaction / Database Records Purge (AML995A)

Comment: @Purpose:

To change the control data area (AM008A) for the purge (program AML993A), instructing it to shutdown.

@Operational Method:

- Change the contents (position 1 - 8) of the data area to cause the program to shutdown.

Process

Hierarchical numeric ID: 1.1.4.1.1.6

Coded name: AML996A

Name: PGM Reorganized Application Database File(s) (AML996A)

Comment: @Purpose:

To maintain the size of the database files by reorganizing the objects, thus reducing amount of time required when accessing the data.

@Operational Method:

- Update the controlling data area with the file name to be reorganized. That way when the job ends in mid stream it can pick up where it let off.

Process

Hierarchical numeric ID: 1.1.4.1.1.7

Coded name: AML997A

Name: PGM Archive Transaction File Data (AML997A)

Comment: @Purpose:

Based on a practical experience of the Application Support Staff, data within certain transaction files are archived for trouble shooting purpose.

@Operational Method:

- Create the archive files when they don't exist for the host machine running the program.

- Each file follows the same archiving procedure:
 - Second month rolled to temporary Third month
 - First month rolled to Second month
 - Current month rolled to First month
 - Empty Third month inserted as Current month

ARMS Process Report

@Note: Presently the data is kept for two months.

Process

Hierarchical numeric ID: 1.1.4.1.2

Coded name:

Name: BAT Startup ARMS Environment

Comment: @Definition: This activity reorganizes the transactions work files, data queues, archives needed information and prepares the environment for starting the ARMS application.

@Notes: The ARMS rejected transactions report is also generated by this activity.

Process

Hierarchical numeric ID: 1.1.4.1.2.1

Coded name: CLL813

Name: PGM Delete and Rebuild Data Queues (CLL813)

Comment: @Purpose:

To delete and rebuild data queues. This is an IBM recommended procedure to improve data queue performance.

@Operational Method:

- CALL CLL813HIST to archive file, AMSET.
- CALL AM0038V1 to save data queue entries (for all data queues except DQAM62V1) and associated data to work files for restoring at a later time
- CALL AM0037V1 to save AMPACK (ARMS Transactions ready to be sent) records that have not been processed to a work file for restoring at a later time
- Read records from ANDQINFO file in a loop and for each data queue in the file, perform the following steps:
 - i) IF data queue name is DQAM62V1 and the machine is NOT DEV, skip this record and read the next record ELSE continue
 - ii) IF the associated data queue file name is blank, skip this record and read the next record ELSE continue
 - iii) IF the machine is RARMS and "RARMS Only" flag is 'Y'es, OR IF the machine is DEV OR IF the machine is NOT RARMS AND the RARMS only flag is NOT 'Y'es
 - IF the data queue name is blank, reorganize the data queue file
 - IF the VAN is not specified OR IF the VAN is specified AND this is the weekly run, clear the data queue file.
 - IF the file is in use, message ARMS On Call
- IF on the DEV machine, CALL CLL814 (as it is not submitted by ROBOT on DEV). On DEV CLL813 is called by CLL991.

Process

Hierarchical numeric ID: 1.1.4.1.2.2

Coded name: CLL813HIST

Name: PGM Backup ARMS Critical Files (CLL813HIST)

ARMS Process Report

Comment: @Purpose:

To backup files ANDIST, AMSET and AM0060LG.

@Operational Method:

- On the DEV machine, archive all three files with the *REPLACE option on the CPYF command to keep the archived file size small.

- On the RARMS machine, archive the set file, AMSET with the *ADD option to append to the previous archive

- On the distributed machine, archive the files ANDIST and AM0060LG to ANDIST_HST and AM0060SV respectively with the *ADD option to append to the previous archives.

Process

Hierarchical numeric ID: 1.1.4.1.2.4

Coded name: AM0038

Name: PGM Save / Restore Data Queue Entries and Associated Data (AM0038)

Comment: @Purpose:

To save the data queue's data to a work file and then delete and recreate the data queue to improve its performance.

@Operational Method: Three parameters are passed to this program: the processing code ('S'ave or 'R'estore), the data queue name and the run type ('D'aily or 'W'eekly).

For 'S'ave processing do the following:

- If the data queue name parameter is *ALL, read file, ANDQINFO in a loop Else read the file with the data queue name as the key and for each data queue that meets the following conditions :

is not DQAM62V1

is not blanks,

the VAN id is blanks OR

the VAN id is not blanks and the run type is 'W'eekly,

the machine is RARMS and the "RARMS Only" flag is 'Y'es,

the machine is 'DEV'

the machine is distributed and the RARMS Only flag is NOT 'Y'es;

Do the following:

IF the key length is greater than zero

receive all keys greater than blank

ELSE

simply receive each entry sequentially.

For each entry except the *DOWN, *STOP and *WAIT entries, write the data queue key and data to a work file until all entries have been received.

For each entry that has associated records in a database file, write the records to a work file.

Then delete the data queue and recreate it.

For 'R'estore processing:

ARMS Process Report

- If the data queue name parameter is *ALL, read the work file in a loop ELSE read the work file with the data queue name as the key and for each data queue entry found in this file,
i) Read the file, ANDQINFO to get the data queue information
ii) Send the queue entry to the corresponding data queue via one of the two subroutines, one for keyed and one for non-keyed data queues.
iii) Write any associated database records in the work file back to the appropriate database file.

For any chain failing in the above, CALL AMPSSR and message ARMS On Call.

Process

Hierarchical numeric ID: 1.1.4.1.2.5

Coded name: AM0037V1

Name: PGM Backup Unprocessed Transactions in Package file AMPACK (AM0037V1)

Comment: @Purpose:

To save unprocessed AMPACK records.

@Operational Method:

Job L813 creates a work file, L813A by sorting AMPACK by company id.

Read previous L813A starting from the last record and save all 'IN'voice and 'OF'fice batch transactions that have not been sent.

Any company that has a format 'APPD02' record in AMPACK has already been processed and so do not save that company's data. Look for *EOBD in the data portion and the format name 'APPD02'. This format is written by program AM0140 (called in CLL811) for each company that the batch transmission is sent to.

Process

Hierarchical numeric ID: 1.1.4.1.2.6

Coded name: CLL810

Name: PGM ARMS Startup (CLL810)

Comment: @Purpose:

To start up the ARMS application system programs that make the ARMS system handle transactions on a continuous basis until a shutdown is initiated.

@Operational Method:

- Call program CC0006 to reset the callback control status flags from "OK" to "DO" in CCEXTCTL Claims Connection Callback file.

- Reset the Number of Transmission Envelope Validation NEP jobs currently active to zero.

-Allocate file AMXREF to determine if there is lock on this file. If there is no lock DeAllocate file and continue. If the file is locked and not available delay job and try again. If on the third attempt the file is still locked page the ARMS oncall.

- Submit the proper Never-Ending Program (NEP) batch jobs to the ARMS Job queue for the centralized ARMS external transaction-based ("real-time") communications (separate sending and receiving jobs for each specific connection),

ARMS Process Report

centralized ARMS processing, centralized internal host platform distribution communications, distributed ARMS processing, distributed rental interface and distributed rental extraction transaction set generation processes. (This is based on the current host system name that this startup processing is being executed.)

@Notes:

This processing is initiated daily around 3:30am. Shutdown: 11:50pm
(all times for Central Time Zone)

@Files (CRUD)

none

Process

Hierarchical numeric ID: 1.1.4.1.2.8

Coded name: CLL817

Name: PGM Copy / Print Rejected Transactions Report & Start Subsystem (CLL817)

Comment: @Purpose:

To consolidate all the transactions rejected by ARMS edits and generate an rejected transactions report after archiving the error log file, AMERRLOG. Also, start the ARMS subsystem if not already started.

@Operational Method:

- IF on the distributed machine, copy file AMERRLOG to the same file on the ARMS host via DDM and then clear the file. If DDM link is unavailable, message on call staff.
- IF on the ARMS host, archive consolidated file AMERRLOG and generate rejected transactions report by calling program AM5000. Reorganize file AMERRLOG to remove deleted records.
- Start the ARMS subsystem if not already started.

Process

Hierarchical numeric ID: 1.1.4.1.2.9

Coded name: AM5000

Name: PGM Print ARMS Error Log Report (AM5000)

Comment: @Purpose:

To generate the report for all the inbound and outbound transactions that were rejected by ARMS edits.

@Operational Method:

- Read file AMERRLOG sequentially. The file is keyed by Company Id, Transmission Control Id, Group Control Id, Group Type and Customer Transaction Id.
- For every record in this file, write the header and all error codes and associated detail. For error codes 68 (Invalid Vendor Transaction Id) and 82 (Request Rejected : can not identify transaction) on an Authorization (AT) group type, retrieve the SET 9formats beginning with "SET") and Renter (formats beginning with "RNT") detail formats from the received transactions log, AMTRNLOG and write them to the detail portion of the report.

Process

Hierarchical numeric ID: 1.1.4.1.2.11
 Coded name: AM0032V1
 Name: PGM Write Saved AMPACK records to AMPACK (AM0032V1)
 Comment: @Purpose:

To restore the saved AMPACK records.

@Operational Method:

This program starts at the bottom of the work file where AMPACK records were saved and does a read prior to restore the data to the file, AMPACK in the proper sequence.

Process

Hierarchical numeric ID: 1.1.4.1.2.12
 Coded name: AML002A
 Name: PGM File Transfer and Archive RMS ARMS Trading Partner Insurance Company
 Transaction File (AML002A)
 Comment: @Purpose:

To transfer the daily RMS (Rental Management System) transaction file record on the centralized ARMS host platform to the RMS interface file on the RMS centralized host platform.

@Operational Method:

This program is submitted to run as a batch job on the centralized ARMS host platform after the CLL814 program batch job has completed and before the CLL810 program batch job has been submitted. The program will attempt to allocate exclusively and if successful, then copy the ARMS RMS Daily Transaction file (AM002P) to an FTP (File Transfer Protocol) remote site and an archive data set.

Retrieve the current platform (machine) name.

IF the platform name is not 'RARMS' (centralized ARMS host), or 'CENTRAL' (centralized RMS host), or 'DEV' (IS development host), then end this program/job.

Retrieve the current executing job's SUBSYSTEM DESCRIPTION name, JOBNAME and JOBDAT.

Clear the RMS staging area file (AM002PFTP).

IF the current platform is RARMS or DEV, then do the following:

.. Check for an empty RMS Daily Transactions (AM002P) file.
 .. IF the RMS Daily Transactions (AM002P) file is empty (no records), then end program/job.
 .. ELSE, Attempt to Allocate file AM002P to prevent all other access.
IF unable to allocate exclusively the AM002P file, then end the program and send a corresponding Enterprise Message System message to the ARMS On-call staff along with a pager message to the ARMS Primary and Backup On-Call pagers indicating "AM002P File is in use".
 .. AND: Copy the AM002P file to the currently empty RMS interface staging area file (AM002PFTP) sequentially from the beginning of the file.
 .. AND: Clear and Deallocate the AM002P file, making it reusable.

ARMS Process Report

IF the current platform is CENTRAL or DEV, then do the following:

..IF RARMS is remotely unavailable ("platform or intersystem communications link is down"), then send a corresponding message of the remote communications failure to the current platform's system operator (QSYSOPR) and resubmit this job on hold for execution later on this platform, using the previously retrieved JOBNAME, JOBDATE and SUBSYSTEM DESCRIPTION name as the JOB QUEUE name and then end this program/job.

..ELSE (RARMS platform is remotely available), then do the following:

....Perform a "copy-pull" with replacement by copying the records sequentially from the AM002PFTP file on centralized RARMS platform to the current platform's AM002PFTP file.

....Check for an empty RMS Daily Transactions Interface Staging Area (AM002PFTP) file.

....IF the RMS Daily Transactions Interface Staging Area (AM002PFTP) file is empty (no records), then end program/job.

....ELSE, do the following:

.....Retrieve the 4 character day of week abbreviation (current OS/400 system value) (Example: 'THUR', 'WED ', 'TUES')

.....Append the first 3 character of the 4-character day of week to the 'AM002P' file name to derive the correct archive file for each day of the week.

.....Copy (sequentially with replace) the file to be transferred via FTP to the derived archive file for today's day of the week.

.....Retrieve the AM007A data area to get the Local Area Network FTP site name for RMS download. (Example: 'FTPCORP01')

.....Execute the FTP (File Transfer Protocol for AS/400 to Personal Computer) for downloading using the QXTXSRCT Text Source File's AMRMSFTPO member for the File Transfer Input Specifications to transfer/copy with append the records on the AS/400 platform in the RMS Daily Transaction Interface Staging Area (AM002PFTP) file into the AM002P.TXT file in the "\rms\" directory on the FTPCORP01 LAN server's public drive.

@Files: (CRUD)

AM002P (-R-D)

AM002PFTP (CR--)

AM002P.TXT (C---) (in the \rms directory of the FTPCORP01 specified drive of the LAN server's public drive)

@Improvement Opportunities:

Change this process from a batch process to a real-time process to send/push any qualified transaction into the RMS database files.

Process

Hierarchical numeric ID: 1.1.4.1.2.13

Coded name: CLL814

Name: PGM Restore Unprocessed Transactions (CLL814)

Comment: @Purpose:

To restore the data queue entries and associated file data that was saved in job L813.

@Operational Method:

The run type 'D'aily or 'W'weekly is passed to this program as a parameter.

- CALL AM0032V1 to restore file AMPACK
- CALL AM0038V1 with the processing code set to 'R'estore to restore the data queues (these data queues contain the key and the data in the data queue itself e.g. DQAM60V1)
- Save the work files and then clear them

Process

Hierarchical numeric ID: 1.1.4.1.3

Coded name:

Name: BAT Shutdown ARMS Environment

Comment: @Definition: This activity shuts down all the ARMS never ending programs that constitute the main ARMS transaction processing stream.

@Notes: The transactions processing stream consists of the following:

- Translators (AM10)
- Host Processing (AM20/21-AM46)
- Distributed Sending (AM100-AM101) from host
- Distributed Inbound Processing (AM61-EC(AT,EX,CN,CM,CC))
- Distributed Outbound Processing (AM60-EC(AC,RA,RE,RC,RN,VM,TR))
- Distributed CC Inbound Processing (AM61-CC(AT,EX,CN,CC))
- Distributed CC Outbound Processing (AM60-CC(AC,RA,RE,RC,RN,IN,TR))
- Distributed Receiving (AM105-AM106) to host
- Host Sending (AM120)

The receiver jobs remain active 24 X 7 hours except when a roll swap is done to switch all processing to the backup ARMS host machine.

The first shutdown entry is sent to each of the translators running (via program AMSTOP2X which is called from CLL812 which is a scheduled job) so that all input to the stream can be stopped. The translator decrements data area DASDV1, passes shutdown to AM20 and ends itself. AM20 monitors data area DASDV1 and when it becomes zero, it passes the shutdown to AM21 and ends itself. AM21 in turn passes the entry to AM25 and ends itself. AM25 then reads data area AM001A to get the number of computer platforms to shutdown and sets on an internal flag to indicate to itself that one shutdown entry has already been received. After this AM25 passes the entry to AM30 and so on up to AM46. These programs do not end with the first shutdown entry in order that they be able to process any data coming from the distributed computer platforms.

AM46 then passes the entry to each of the distributed sending jobs (AM100) so that the distributed streams can be stopped. AM100 passes the shutdown to AM101 with a detach signal. When AM61 receives the shutdown entry, it reads file AMVRGTST, keyed by group type and source type (e.g. ATB, ATC etc.) to shutdown each of the inbound format generators and then sends the shutdown to AM60 which in turn stops the outbound format generators before ending itself. Each of the format generators, AM61 and AM60 send a shutdown entry to AM106 before ending. AM106 maintains a count of all applicable records in AMVRGTST and when that count is met, it passes the shutdown entry to AM105 along with a detach signal before ending.

Note that AM101 and AM105 can distinguish between a single machine shutdown and a complete system shutdown. For e.g., AM105 will not pass the shutdown to AM25 if it was a single machine shutdown.

ARMS Process Report

AM25 then increments the data area, AM002A, which contains the count of computer platforms that have shutdown. When this count equals the machine count in data area AM001A (manually maintained), AM25 knows that the required number of computer platforms are down and now it can end the remaining host processing stream that is still up by passing on the entry until it reaches AM46.

AM46 then passes the entry to AM120 which stops each of the senders based on the file ARMSPR3 and then ends itself.

Process

Hierarchical numeric ID: 1.1.4.1.3.1
Coded name: CLL812
Name: PGM Stop ARMS Never-Ending Programs (CLL812)
Comment: @Purpose:

To initiate the automated and orderly process for ARMS system shutdown in order to completely process any transactions that have already been started and then shutdown the whole system.

@Operational Method:

- Send Shutdown Request transaction to Trading Partners so profiled to be able to receive a shutdown request.
- After five (5) minutes, then end the remaining communication jobs via an ENDJOB command.
- Send a Shutdown Transaction to each connect-specific receiver output data queue.

@Notes:

Upon the receipt of the first shutdown transaction by the centralized validation activity component programs (AM0025, AM0030 and AM0040) and the centralized database update activity component process program (AM0046), the first shutdown transaction is indicated and the shutdown transaction is passed along to the next program. After receiving the second shutdown transaction, the shutdown transaction is passed to the next program and causes the current program to end in an orderly manner.

Process

Hierarchical numeric ID: 1.1.4.1.3.2
Coded name: AMSTOP2X
Name: PGM Stop All Active Translator jobs (AMSTOP2X)
Comment: @Purpose:

To shutdown all translator and any receive mapper jobs running in the ARMS subsystem.

@Operational Method:

- Retrieve all companies and technical connection information for each company from the profile file, ARMSPR3 and send a shutdown entry to the appropriate receive data queue to shutdown the translator (AM10) job for each company or VAN as the case maybe.

ARMS Process Report

- Then send shutdown entry to the ARMS Tracking Partner Insurance Company receive mapper job. This is hard coded in this program.

Process

Hierarchical numeric ID: 1.1.4.1.3.3
Coded name: RSWAPSTR
Name: PGM End ARMS jobs if necessary (RSWAPSTR)
Comment: @Purpose:

To shutdown the ARMS processing stream on the current machine as part of the preparation for the role reversal between the current and the backup machine.

@Operational Method:

- Check the ARMS and A4BAT subsystems for jobs to see if jobs L812 and LAMSTOP2R have run.

If there are jobs running in either subsystem do the following:

- i) if the time is between 0300 and 2345, submit a job to perform an immediate end of the ARMS processing stream by calling CLL812 with a delay of zero.
- ii) if the time is between 0300 and 2200, end jobs L811 and L749.
- iii) if the time is between 2245 and 2530, call AMSTOP2R with a zero delay to immediately end all the receiver communication jobs.
- iv) once all the above steps have been performed, end the ARMS Subsystem to flush any remaining jobs.

Process

Hierarchical numeric ID: 1.1.4.1.3.4
Coded name: CLSR01
Name: PGM Role reversal CLP (RARMS1 <-> RARMS2) (CLSR01)
Comment: @Purpose:

To shutdown all processing on the current machine and start the ARMS processing on the backup machine.

@Operational Method:

- This CL is executed by operations when there is a need to perform a role reversal between the current and the backup ARMS machine.

@Notes:

This CL includes a lot of processing for ending several subsystems that may be running on the current machine and the code is maintained by operations.

The source for this CL can be found in the SPLIB/QCLSRC.

Process

Hierarchical numeric ID: 1.1.4.1.3.5
Coded name: AMSTOP2R
Name: PGM Stop All Active Comm. Receive Jobs (AMSTOP2R)

ARMS Process Report

Comment: @Purpose:

To shutdown the communications receiver jobs for all companies and to end the pseudo sender job, A4200RB for the special case of ARMS/400.

@Operational Method:

- Retrieve the profile information to construct the receiver job name for each company and end the job.

- For ARMS/400, send a data queue entry to DQ4200RB to end the A4200RB (ARMS/400 pseudo sender) NEP.

Process

Hierarchical numeric ID: 1.1.4.1.3.6

Coded name: DQ4200RB

Name: DTQ Input for ARMS/400 Application (DQ4200RB)

Comment: @Definition: This is the input data queue to the pseudo sender for ARMS/400, program A4200RB that simulates sending to the ARMS/400 customers.

Process

Hierarchical numeric ID: 1.1.4.1.4

Coded name:

Name: ONL Reroute Unsolicited Authorizations

Comment: @Definition:

To activate the rerouter job for any unavailable machine.

@Operational Method:

When a production machine goes down, the ARMS OnCall staff is notified and then the OnCall staff manually submits this job to reroute or hold any transactions that were meant for that machine.

@Note:

- Runs on RARMS only

Process

Hierarchical numeric ID: 1.1.4.1.4.1

Coded name: AMRREROUTE

Name: PGM Submit ARMS Rerouter (AMRREROUTE)

Comment: @Definition:

This is a batch process triggered manually as a result of a distributed production machine that has been abnormally shut down to allow for certain transactions that would have been routed to that machine to instead be routed to CLAIMS CONNECTION and the remainder of the transactions intended for that machine to be held for later processing.

@Purpose:

To allow Enterprise a means of fulfilling contractual ARMS obligations in event of a distributed machine failure.

ARMS Process Report

@Operational Method:

- Submits the AMREROUT never-ending program with all of the necessary values.

@Files: None

Process

Hierarchical numeric ID: 1.1.4.1.4.2

Coded name: AMREROUT

Name: PGM Reroute Transactions for Unavailable Machine (AMREROUT)

Comment: @Purpose:

To handle transactions that are bound for a machine that is unavailable so that the transactions are not lost. If the transactions are new authorizations, someone needs to attend to them as most trading partners have agreements with Enterprise to respond to customer authorization requests within a specified time interval.

@Operational Method:

IF the received transaction set is an unsolicited authorization add (AT);

- Call AM1010V1 to update the XREF with Machine & Source Id, update the AMAPP record and route to Claims Connection

IF the received transaction set is an authorization change (AT-C) OR a cancellation (CN), and the Source or Machine from the APPD01 doesn't match that found in the XREF;

- update the AMAPP record and route the transaction to Claims

Connection

ELSE

- route to hold data queue.

At SHUTDOWN, reroute all held transactions to the production machine that went down.

@Files:

- AMAPPS (-RU-)

- MACHIDENPF (-R--)

Process

Hierarchical numeric ID: 1.1.4.1.4.3

Coded name: DQAMAPH

Name: DTQ Hold Unavailable Machine Transactions (DQAMAPH)

Comment: @Definition: DQAMAPH is a data queue used during reroute for holding all transactions that are not unsolicited AT-Adds (which are routed to CLAIMS CONNECTION) for the affected machine which is down, and then at reroute shutdown for the that machine these entries are then used as input to allow for their normal processing.

Process

Hierarchical numeric ID: 1.1.4.2

Coded name:

Name: SA Research and Fix Problems

ARMS Process Report

Comment: @Definition: The activity to handle any problems that are reported via a pager, a phone call or an MS01 message. It also involves daily tasks to examine exceptions and fix them or to create issues if they can not be fixed.

Process

Hierarchical numeric ID: 1.1.4.2.1

Coded name:

Name: MAN Respond to On-Call Message

Comment: @Purpose:

To research and determine how to solve a problem that was notified via a pager, a phone call or an MS01 message.

@Operational Method:

- The On-Call staff (primary and backup) are notified in the following situations:

1) ARMS job's program halt or termination via pager message to investigate why and to respond and possibly resubmit job for never-ending program (NEP) execution.

2) ARMS transaction set transmission being rejected for processing by an ARMS NEP job via a pager message and/or via e-mail (currently MS01).

- The primary only On-Call staff is notified by ARMS Rental Management Trading Partner technical contact of their host system availability or connect problems or for transaction research for a perceived problem via pager message (8063). The contact actually dials in (314) 512-ARMS and is appropriately directed via four options :

- 1 - Enterprise Rental Branch Personnel
- 2 - Questions about Vehicle Messaging System
- 3 - Insurance Company Contacts
- 4 - Other

The primary only On-Call staff is also notified by the ARMS Programming Help Desk staff via pager message (8064) of problems where they need further assistance.

- To retrieve the 8063 and 8064 mail box messages, perform the following steps:

- i) Dial phonemail - x2696
- ii) When asked, enter the mailbox number from the pager (8063 or 8064) and the password
- iii) After listening to the message, always save it
- iv) Call the concerned party and always keep them apprised of the status of the investigation.

@Notes:

A top priority is to resubmit/restart any abended ARMS NEP job to ensure transaction data flow throughput between the trading partners and the rental systems. Another high priority is to notify ARMS Technical to contact all affected trading partners of any communication termination occurrences that are estimated to be longer than thirty (30) minutes in duration. Likewise, they are also to be notified of any similar delays of transaction throughput. A lower priority is to research the cause of any transaction being rejected or causing program halt/termination, correct and resubmit the transaction.

Process

Hierarchical numeric ID: 1.1.4.2.2
 Coded name:
 Name: MAN Perform Daily Exception Checks (Checklist)
 Comment: @Purpose:

To use the check list and perform daily tasks that have been identified as routine to make sure that the application processes any exceptional transactions.

@Operational Method:

- If any printed detail lines exist on the ARMS Cross-Reference Discrepancy Reports spooled file in the ARMSRPTS output queue on the RARMS centralized host platform, print the reports. Then research all ARMS activity to-date of the rental transaction via the ARMS Time-Line Inquiry (CCAM16) and the ARMS centralized database report (AM0800). Manually correct the files' records or generate the necessary transaction set for the needed group type.
- Display the RARMS host platforms distribution data queue (DQAMAP1) to determine if there is any old (older than today's date) or hung transaction that has never been sent. Research the associated transaction set's transmission and data queue entry for invalid host system platform ID value. Correct and reprocess if necessary.
- Display the distributed rental systems' host platforms ARMS Application Interface Transaction Set file (AMAPP), the ARMS Format Generation Dispatcher In-Progress Transaction Control File (AMTRNCTL), and all distributed ARMS application system data queues (DQAM6AV1, DQAM6BV1, DQAM6OV1, DQAM61V1, and DQECATV1) to determine if there is any old (older than today's date) or hung transaction that has never been processed. Research the associated transaction set's transmission and data queue entry (along with the existing rental contract transaction) for invalid rental contract IDs values or status. Correct and reprocess if necessary.
- Check the spooled print file archive system's spooled file for the latest J340X Daily Rental Billing Job runs' Billing Exception Report for rejects due to ARMS status code or missing ARMS Cross-Reference file record error message. If any exist, note the rental location and contract ID. Research all ARMS activity to-date of the rental transaction via the ARMS Time-Line Inquiry (CCAM16) and the ARMS centralized database report (AM0800). Manually correct the files' records or generate the necessary transaction set for the needed group type.
- Check if any printed detail lines exist on the query reports from last execution of job AML900 in two spooled files in the ARMSRPTS output queue on the RARMS centralized host platform, print the reports. Then research all ARMS activity to-date of the rental transaction via the ARMS Time-Line Inquiry (CCAM16) and the ARMS centralized database report (AM0800). Manually correct the files' records or generate the necessary transaction set for the needed missing opening "RN" group type.
- Every weekday morning, change the centralized Job Monitor switch (on CENTRAL) for the ARMS On-Call Pager to "YES" so that ARMS On-Call pagers will be immediately sent notification messages of any ARMS job halts or abends.
- Every business weekday afternoon around 4:00pm, except the last business day of the week, change the centralized Job Monitor switch (on CENTRAL) for the ARMS On-Call Pager to "NO" so that ARMS On-Call pagers will NOT be sent

ARMS Process Report

notification messages on any ARMS job halts or abends. These notification pager messages will be sent instead to the Operations On-Site Pager.

Process

Hierarchical numeric ID: 1.1.4.2.3

Coded name:

Name: MAN Research and Resolve Network Problem Ticket

Comment: @Purpose:

To research and resolve and problem for which a network ticket has been created by the technical support center.

@Operational Method:

- Any time a Network Problem Ticket is passed to the Technical Support Center's Programming Help Desk (TSC-PHD) staff require further research and/or suggestions on correcting a deficient transaction situation with a rental transaction that had been processed through ARMS, then they call the ARMS Technical Support Line that generates an "8064" mailbox pager message to indicate the Application Development Department primary on-call person that a new voice mail message has been placed into the "8063" mailbox. The on-call person will listen to the voice mail message and then contact the caller to discover more details that might aid in their research. Then the on-call person (or can pass the task to the backup on-call if the primary has no time available for such a task) does the research into the problem and reports back to the caller the necessary intervention that the on-call person, TSC-PHD person, Rental Systems Help Desk staff, rental user, or trading partner system user must perform to correct the deficient situation.

Process

Hierarchical numeric ID: 1.1.4.3

Coded name:

Name: SA Information System Reports

Comment: @Definition: The activity that prints the various reports provided for in the ARMS application. The reports are as follows:

- Insurance company referral closed rental contracts revenue
- Management report for direct billed invoices
- Month-end ARMS youthful drivers surcharge discount report
- ARMS transaction exception report
- Reservation detail by company utilization management report

Process

Hierarchical numeric ID: 1.1.4.3.1

Coded name: CLJ470

Name: BAT Print Insurance Company Referral Closed Rental Contracts Revenue Reports (CLJ470)

Comment: @Definition: The batch process for printing of various Insurance reports, as specified in the Operational method and run on the Query machine. This process is triggered by TIME.

@Purpose: To report on various ARMS and Rental data in support of the Marketing function.

@Operational Method:

ARMS Process Report

- Print Insurance Referral Report of Gross Revenue
- Print Insurance Referral Report of Income
- Print Bill-To Customer Report by Group
- Print Referral Reports by State (Gross and Net Revenue)
- Print Bill-To Customer Report by State

@Files: (CRUD) (Unless noted, all files used reside on QUERY system)

- OFFDRB (-R--)
- IR470P (-R--) (on CENTRAL system)
- INSRMOYR (-RU-) (on CENTRAL system)
- INSRMOYR (CRU-)
- INSRGGP (-R--) (on CENTRAL system)
- INSRGGP (CR-D)
- INSRPHD (-R--) (on CENTRAL system)
- INSRPHD (CR-D)
- RACCLS (-R--)
- RACCLS01 (-R--)
- RACCLS02 (-R--)
- RACCLS03 (-R--)
- RACCLS04 (-R--)
- RACCLS05 (-R--)
- RACCLS06 (-R--)
- RACCLS07 (-R--)
- RACCLS08 (-R--)
- RACCLS09 (-R--)
- RACCLS10 (-R--)
- RACCLS11 (-R--)
- RACCLS12 (-R--)

@Notes:

These reports are confidential and are used by the Rental Operations - National Marketing Department staff.

Process

Hierarchical numeric ID: 1.1.4.3.2

Coded name:

Name: BAT Transaction Credit Discount

Comment: @Definition: The batch process for printing the month end transaction credit report.

@Purpose: For ARMS Technical Management and Rental Operations National Marketing Department staff to compute ARMS utilization for any transactions credit rebate checks sent to the Rental Management Trading Partner from closed direct billing payment authorized rental contracts.

@Notes:

Report is used by ARMS Technical Management and Rental Operations National Marketing Department staff to compute ARMS usage for any transactions credit rebate checks sent to the Rental Management Trading Partner.

Process

Hierarchical numeric ID: 1.1.4.3.2.1

Coded name: AML968A001

Name: PGM Create Missing OUTQs for Credit Transactions Report (AML968A001)

Comment: @Purpose:

To create missing output queues on the RARMS machine at the beginning of the Credit Transaction Report driver job (CLL968) program so it can complete normally.

These output queues are named: Pgg, PggPC, or PggrPC where "gg" is the Group ID and "r" is Region ID and are created in library QUSRSYS using the CRTOUTQ command with default parameters except for the OUTQ name.

@Operational Method:

Repetitively read through Office Directories Group Master File by GROUP ID (file OFFDG#2).

For every record read, do the following only IF the record's GROUP ID field value is not equal to '76' (Claims Connection) AND the GROUP ID field value is equal to the GROUP'S OUTPUT QUEUE NAME:

- IF the record's Distributed Printing Capable indication field is 'Y' (Yes - a PC printer output queue is located at this Group administration office), then check if that Group's PC output queue name ("PggPC" - where "gg" is the Group ID) exists on this platform. IF it does not exist, then create the missing personal computer (PC) output queue (OUTQs) for this Group Administrative office.

- ELSE (Yes - a PC printer output queue is NOT located at this Group administration office), then check if that Group's normal output queue ("Pgg" - where "gg" is the Group ID) exists on this platform. IF it does not exist, then create the missing normal peripheral output queue (OUTQs) for this Group Administrative office.

- IF the retrieved record's Regionalized indication field is 'Y' (Yes - this Group is administrated by other regional administration offices), then execute the ARMS Retrieve Regions for Office Group (AML968A002) program, passing two parameters. This first is the input parameter of the current record's GROUP ID. The second parameter is the blanked 132 element, 1 character output parameter of a Region ID array of all regions that are part of a Group office. Then, looping through each non-blank array element, check if the Regional Office's of the Group personal computer output queue name ("PggrPC" - where "gg" is the Group ID and "r" is the Region ID) exists on this platform. IF it does not exist, then create the missing personal computer (PC) output queue (OUTQs) for this Group's Regional Administrative office.

@Files: (CRUD)

OFFDRGRPPF (-R--) Office Directories Group Master File (by Group ID, using logical access path OFFDG#2).

Process

Hierarchical numeric ID: 1.1.4.3.2.2

ARMS Process Report

Coded name: AM1081V1

Name: PGM Print Detail and Summary Transaction Credit Reports (AM1081V1)

Comment: @Purpose:

To print ARMS Transaction Credit summary and detail reports by Trading Partner.

@Operational Method:

Positions and reads the ARMS Transaction Credit Transaction History (AMTRNCR) file records whose Year and Month Approved for Payment/Rental Closed correspond to the previous month/year from the current system date.

This physical file's indexed/keyed access path is sequenced by:

Year Approved for Payment or Rental Closed
Month Approved for Payment or Rental Closed
Trading Partner Profile ID
Group ID
Region ID
Branch ID
Ticket ID Number (of closed rental contract)

Regionalized the Transaction Credit summary report. This report matches the report that is sent to the Group output queues except that it includes total information for branches within a region and total information for the region itself, but does not include Group total information.

A total line to the Transaction Credit Detail report exist just for ARMS Trading Partner Insurance Company. The total will be the sum of the transaction credit and the utilization fee.

(If there is a Trading Partner with no utilization fee, this total line will not print.)

The hierarchy of breaks are:

Trading Partner Profile
Group
Region
Branch

Accumulate Branch, Region (if applicable), Group and Trading Partner Totals for:

Total Count of (Rental Contracts) Tickets Closed (Managed via ARMS)
Total Insurance Charged Extended Daily Rental Rate Revenue Amount of
(Rental Contracts) Tickets Closed (Managed via ARMS)
Total Transaction Credit (Discount) Amount (to be rebated)
Total (ARMS) Estimated Utilization Fee Amount

Print detail report lines (used for auditing when necessary)

Print report break totals on detail and summary report printer files.

Page breaks on: Trading Partner Profile ID or GROUP ID changes.

@Files: (CRUD)

AMTRNCR (-R--) ARMS Transaction Credit transaction history file

ARMSPR7 (-R--) ARMS Trading Partner's Transaction Discount

Volume Break Control Levels

ARMS Process Report

ARMSPR2 (-R--) ARMS Trading Partner's Customer Address Data

OFFDRGRPPF (-R--) Office Directories Group Master File, by GROUP ID
(using the OFFDG#2 logical access path)

QSYSPRT (C---) System Default PRINTER FILE for Detail Report by
GROUP ID, REGION ID, and BRANCH ID

QSYSPRT2 (C---) System Default PRINTER FILE for Summary Report by
GROUP ID, REGION ID, and BRANCH ID

QSYSPRT3 (C---) System Default PRINTER FILE for Detail Report by
GROUP ID, REGION ID, and BRANCH ID for the Group Administration Office Output Queue

QSYSPRT4 (C---) System Default PRINTER FILE for Detail Report by
GROUP ID, REGION ID, and BRANCH ID for the Regional Administration Office Output
Queue

@Embedded Data/Constants:

Program Constants:

'GROUP SUMMARY - FINAL TOTALS'
'UTIL FEE \$'
'UTIL FEE'
'UTIL %'
'TOTAL'

** CMD - Commands To Be Executed

OVRPRTF FILE(QSYSPRT3) OUTQ(QPRINT)
OVRPRTF FILE(QSYSPRT3) OUTQ(PggPC)
OVRPRTF FILE(QSYSPRT3) OUTQ(Pgg)
DLTOVR FILE(QSYSPRT3)
OVRPRTF FILE(QSYSPRT4) OUTQ(PggrPC)
DLTOVR FILE(QSYSPRT4)

** Month Names for report column headings.

JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
JULY
AUGUST
SEPTEMBER
OCTOBER
NOVEMBER
DECEMBER

@Notes:

Utilization fee is the amount paid by Enterprise to the customer for
sending rentals using ARMS as specified in the contract with the company. For ARMS
TRADING PARTNER INSURANCE COMPANY, currently they are paid a utilization fee of 1.8 %
of any revenue generated via ARMS that is over 55 million.

The report is broken down by group and branch so that corporate can
recover these fees from the appropriate group-branch based on the revenue that each
branch generated via ARMS.

ARMS Process Report

@Improvement Opportunities:

Convert program from OPM to ILE.

Remove the compile-time array for the months' names and replace with the logic to execute a built-in function utility or service program that you can pass the date and the month name will be returned as output.

After having printed and saved the spooled printer files, if specified, then move the spooled files to another specific output queue for interception for archival by the Broderick Spool Archival utility software system. This would prevent the need for any reruns for the sake of a reprint.

Move the logic that provides the month and year values to select records for reporting from this program to the calling program, CLL968, and pass the derived last month's month and year as input parameters to this program. This would enable rerunning this program for a different past month that data was still retained on-line.

Optimum: Capture and retain all necessary information in a revised closed rental contracts file and use this type report over any closed rental contracts files by a past month and year for any rerunning.

Process

Hierarchical numeric ID: 1.1.4.3.2.3

Coded name: AML968A002

Name: PGM Retrieve Regions for Office Group (AML968A002)

Comment: @Purpose:

To return all associated Region IDs for a given Group ID so that the AML968A001 program can check if the regional administrative office's personal computer output queue exists on the current platform and if not existing, create them prior to the execution of the Transaction Credit Report (AM1080V1) program.

@Operational Method:

This program accepts two parameters, the first is the input 2 character GROUP ID. The second is the 132 element, 1 character each output REGION ID LIST array.

Starting at the first record in the Office Directories Branch Office Master File for the passed GROUP ID, read every record for this passed GROUP ID value until it changes or is end-of-file, do the following for each associated record retrieved:

- IF the REGION ID is NOT blank, then load the next available REGION ID LIST array element.

End the program and return the loaded REGION ID LIST array to the calling program.

@Files: (CRUD)

OFFDRB (-R--) Office Directories Branch Office Master File, by Group ID, Region ID, Area ID & BranchID (via OFFDRB7 logical access path).

ARMS Process Report

@Improvement Opportunities:

Condition loading the array only after checking to ensure that the retrieved OFFDRB7 file's non-blank REGION ID field value is different than the last loaded value, since the file access sequence is in GROUP ID and REGION ID ascending order. This will avoid loading the array with repeating values. Likewise, the 132 element REGION ID LIST array would not need the maximum of 132 and could be reduced to a smaller size such as the 94 characters (upper and lower case alphabetic, numeric, and special characters) that exist on the computer terminal's keyboard.

Process

Hierarchical numeric ID: 1.1.4.3.2.5
Coded name: CLL968
Name: PGM Drive Transaction Credit Discount Report (CLL968)
Comment: @Purpose:

To print summary and detail reports, by Rental Management Trading Partner companies, of ARMS authorized transactions count and amounts that became closed authorized rental contracts and include the amount electronically invoice billed, excluding government taxes and surcharges.

These reports assist ARMS Technical Management and Rental Operations National Marketing Department staff to compute amount of ARMS utilization by Rental Management Trading Partner so that appropriate rebate checks may be sent to the trading partner.

@Operational Method:

Determine the current emulated or real hardware platform upon this program is executing.

IF the current hardware platform is "RARMS", then do the following:

Override the default system printer files QSYSPT and QSYSPT2 to have the attributes of COPIES set to 2.

Execute the ARMS Create Missing Output Queues for Credit Transactions Reports (AML001A001) program.

Execute the ARMS Print Detail and Summary Transaction Credit Reports (AM1080V1) program.

@Notes:

Submitted for execution from ROBOT/Scheduler as "L968C05" every month on the fifth calendar day of the month only on the RARMS system.

Report is used by ARMS Technical Management and Rental Operations National Marketing Department staff to compute ARMS usage for any transactions credit rebate checks sent to the Rental Management Trading Partner.

Process

Hierarchical numeric ID: 1.1.4.3.3
Coded name: CLL480
Name: BAT Print Management Report for Direct Billed Invoices (CLL480)
Comment: @Purpose:

ARMS Process Report

To report on various ARMS and Rental data relating to direct billed invoices in support of the Marketing function.

@Operational Method:

- Prints or sends to National Marketing representative (via FTP) the Rental Management Trading Partner Company Generic Insurance Direct Billed Report, for Selected Customers or a set of standard Branch Claims Office Customer Numbers.

@Files: (CRUD)

- INSRL480 (-R--)
- IR002P (CR-D)
- IR003P (CR-D)
- OFFDRB (-R--)
- RACCLSC (-R--)
- RACCLS01 (-R--)
- RACCLS02 (-R--)
- RACCLS03 (-R--)
- RACCLS04 (-R--)
- RACCLS05 (-R--)
- RACCLS06 (-R--)
- RACCLS07 (-R--)
- RACCLS08 (-R--)
- RACCLS09 (-R--)
- RACCLS10 (-R--)
- RACCLS11 (-R--)
- RACCLS12 (-R--)

@Notes:

These reports are confidential and are used by the Rental Operations - National Marketing Department staff.

FTP process - Reports being downloaded are written to members in file IR002P. The names of these members are written as records in file IR003P. Program IR_018M is then called by CLL480 to send the data to the internet location. IR_018M is made of 4 modules: CL IR_018M001 which calls IR_018M002 (BUILDER) to build the FTP copy commands (one per member) and to separate the members in IR002P into files in Library QTEMP (module IR_018M004 does this piece) and then calls IR_018M003 (CHECKER) to read the FTP log to see if errors occurred. A batch job then runs on the internet location which checks for data in the appropriate directory and, if data exists, creates an executable that appears on Cindi Loomis of National Marketing's PC desktop. This batch job runs hourly Monday through Friday during normal business hours. There will be one executable for each downloaded file. When Cindi selects these, they are copied to diskette.

Process

Hierarchical numeric ID: 1.1.4.3.4

Coded name:

Name: BAT Youthful Drivers Surcharge Discount Report

Comment: @Purpose: The batch process of generating and printing the month end management report that deals with youthful drivers surcharge discounts

@Notes:

Distributed to Mary Mahoney, Rental Operations - National Marketing Department Assistant Vice President

Process

Hierarchical numeric ID: 1.1.4.3.4.1

Coded name: AM1087V1

Name: PGM Generate Youthful Drivers Surcharge Discount Report (AM1087V1)

Comment: @Purpose:

To generate youthful drivers surcharge discount report.

@Operational Method:

- Print ARMS Youthful Driver Approved Surcharge Discount Detail Report by Group/Branch.
- Print ARMS Youthful Driver Approved Surcharge Discount Summary Report by Group.
- Print Exception Report of ARMS Transaction Credit Discount Records with closed rental contract (ticket) file record not found.

@Notes:

This report is used by National Marketing to determine discount benefit levels. Essentially, if the underage driver surcharge is not paid by the trading partner, it a discount for them and this report lists the actual discount amounts by trading partner.

This report is currently hard coded to be produced only for ARMS TRADING PARTNER INSURANCE COMPANY. The surcharge amount used for calculating the discount is \$5.00 per day per qualified rental. Any rental in AMTRNCR is qualified. The number of days for the rental is calculated from information retrieved from the closed ticket files on the QUERY machine.

Process

Hierarchical numeric ID: 1.1.4.3.4.2

Coded name: CLL987

Name: PGM Setup Youthful Drivers Surcharge Discount Report (CLL987)

Comment: @Purpose:

To prepare the environment and run the program to generate the youthful drivers surcharge discount report.

@Operational Method:

- 1) Retrieve the current date and construct the REPORT DATE in format CYYMMDD using the previous month with the appropriate century and year.
- 2) Retrieve the machine name.
 - If currently running on centralized ARMS host,
 - retrieve the QUERY machine status
 - If DDM is not available,
 - Submit the job so that when the machine is available, the job can be released by operations

Else
Return.

- 3) Change the printer file attributes so that 2 copies are generated for each report.
- 4) CALL program AML1087V1 with the REPORT DATE as the parameter.
- 5) Return

Process

Hierarchical numeric ID: 1.1.4.3.5

Coded name:

Name: BAT Cross-Reference Exception Reports

Comment: @Definition: This is a batch process that generates a report of the discrepancies that exist in the ARMS XREF files between the centralized machine and the distributed computer platforms.

@Purpose: To maintain and keep the centralized and distributed XREFS in synch.

@Notes:

Reports used by the ARMS On-Call staff on daily basis for research and possible manual intervention to fix or resynchronize the centralized and distributed ARMS Cross-Reference database files.

Process

Hierarchical numeric ID: 1.1.4.3.5.1

Coded name: CLL749

Name: PGM Consolidate Transaction Data (CLL749)

Comment: @Purpose:

To set up the environment and run the program to generate the cross-reference discrepancy report.

@Operational Method:

- When on the centralized host machine:
 - 1) Check all distributed computer platforms that are ARMS active - defined in system list ACTVARMS, for DDM availability.
If all computer platforms are not available end the program.
 - 2) For each distributed host system that is included in ACTVARMS system list,
copy the cross-reference file to the centralized host machine, where it is consolidated into ZAMXREF.
- When on the DEVELOPMENT machine:
 - 1) For each simulated (dummy DDM) distributed file that is ARMS active - defined in system list ACTVARMS,
copy the cross-reference file to the consolidated file, ZAMXREF.
- If not on either of the above mentioned host computer platforms, send program message 'YOU ARE NOT ON THE DEV OR RARMS MACHINE, PROGRAM CLL749 ENDING'.

ARMS Process Report

- Check if there are any records accumulated in ZAMXREF.
If there are no records,
send message 'THERE WERE NO RECORDS
TO PROCESS, PROGRAM CLL749 ENDING'
Else
Override to print to output queue ARMSRPTS
CALL program AM0749 to compare the cross references and generate
the discrepancy report.
- Return

Process

Hierarchical numeric ID: 1.1.4.3.5.9

Coded name:

Name: RPT Exception of Discrepancies Between Centralized Cross-reference and Distributed

Comment: @Purpose:

Report use by the Application Support Staff for the sake of keeping the database in synch.

@Report Distribution:

No automatic distribution. The staff on call has to go to the output queue ARMSRPTS to retrieve the report.

Process

Hierarchical numeric ID: 1.1.4.3.5.11

Coded name: AM0749

Name: PGM Print Discrepancies Between Centralized Cross-reference and Distributed (AM0749)

Comment: @Definition:

This is a batch process that generates the ON-CALL report of discrepancies that exist in the ARMS XREF files between the centralized machine and the distributed machines.

@Purpose:

To maintain and keep the centralized and distributed XREFS in synch.

@Operational Method:

- Process the data in the consolidate file against the data in the centralilzed file. Each record will fall into one of the following categories:
 - 1) Record(s) exist on the distributed system only.
 - 2) Record(s) exist on the centralized system only and no error associated with the transaction.
 - 3) Record(s) exist on both centralized and distributed system, but data not in synch.
- Print the three reports.

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.4.3.7

Coded name:

Name: BAT Added 'I'nsurance Customer

Comment: @Definition: This activity generates the report listing all the insurance type customers that were added to file CUSTMAST using customer maintenance program AACM07 since the report was last generated.

Process

Hierarchical numeric ID: 1.1.4.3.7.1

Coded name: CLL010

Name: PGM Generate Report of Added Insurance Customer (CLL010)

Comment: @Purpose:

To set up the environment and invoke the program to generate a report listing all the insurance type customers that were added to file CUSTMAST using customer maintenance program AACM07 since the report was last generated.

@Operational Method:

- Consolidate the data (file CIADDS) from all distributed computer platforms on to the financial application system's host platform and then generate a report from the consolidated data.

Process

Hierarchical numeric ID: 1.1.4.3.7.8

Coded name:

Name: RPT Added 'I'nsurance Customer

Comment: @Purpose:

To inform the user of all the insurance type customers that were added in the last week using program AACM07 (ECS Customer Maintenance).

@Report Distribution:

Cindy Loomis

Process

Hierarchical numeric ID: 1.1.4.3.7.10

Coded name: IR0100

Name: PGM Print Weekly Report of Added 'I'nsurance Customer (IR0100)

Comment: @Purpose:

To generate the report of all the insurance type customers that were added in the last week using program AACM07 (ECS Customer Maintenance).

@Operational Method:

1) Read through file LD010 (work file with consolidated data from file CIADDS on each machine) and write one detail line for each customer in this file to generate the report.

Process

Hierarchical numeric ID: 1.1.4.3.12

Coded name:

Name: BAT Missing Open Rental Notification

Comment: @Definition: A batch process that generates a discrepancy report of authorization and associated ticket that are linked yet not completely updated or in synch with the transaction cross-reference file.

@Purpose: To inform the application staff of contracts that are linked to authorizations yet do not contain the authorization information or are not in sync with the transaction cross-reference file.

@When all the pieces are in place the program is capable of generating the necessary request to ARMS to generate the opening rental notification.

Process

Hierarchical numeric ID: 1.1.4.3.12.1

Coded name: AML901A

Name: PGM Submit Report Program (AML901A)

Comment: @Purpose:

To submits the report program to generate missing RN report.

@Operational Method:

- Submit the job to call program AMZ001A to detect the missing opening RNs and generate the report for the on call staff to investigate.

@Notes:

- This program is called by a scheduled job (AML901A) in ROBOT at 2100 hrs.

Process

Hierarchical numeric ID: 1.1.4.3.12.2

Coded name: AMZ001A

Name: PGM Detect / Generate Missing Open Rental Notification Report (AMZ001A)

Comment: @Definition:

A batch process that generates a discrepancy report of authorization and associated ticket that are linked yet not completely updated or in sync with the transaction cross-reference file.

@Purpose:

To inform the application staff of contracts that are linked to authorizations yet do not contain the authorization information or are not in sync with the transaction cross-reference file.

@Operational Method:

- For every ticket in the file RACMAST that is completely open i.e. CONTRACT DATE has a valid date, STATUS is blank, RMRNO (reservation number) is not blank and TYPE is 'I':

ARMS Process Report

1) Retrieve the reservation master file RACBRMST with the reservation number as the key and compare the ticket numbers in the two files:

CASE 1: If the ticket number in RACBRMST is = 999999, then ignore this ticket and go to the next ticket.

CASE 2: If the ticket number in RACBRMST is not equal to ticket number in RACMAST, then generate message "Ticket linked to Res, but Res points to different Ticket."

CASE 3: If the ticket numbers in the two files match, then:

a) Retrieve supplemental ticket file RACSMAST.

- If no record found,
then go to next ticket

Else

- If ARMS Flag in RACSMAST (RMARMS)
and RACBRMST (BRARMS) both are blank,
then go to the next ticket

Else

- If the flags are different, then report message
"Ticket and Res linked to
each other, however ARMS Flags are different"

Else

call cross-reference retrieval program,

- If call fails, report message

"cross-reference retrieval failed"

Else

use the following decision table.

- C1 : Xref found

C2: Xref not found

C3: Xref status = 'O'

C4: Xref status <> 'O' and <> 'R'

C5: Xref status = 'R'

C6: Transaction exists in AMTRNCTL

C7: Transaction does not exist in AMTRNCTL

If C2; report message "No cross-reference found"

If C1 and C3, go to next ticket

If C1 and C4, report message "Cross-reference status not R"

If C1, C5 and C6, report message "Record exists in file AMTRNCTL"

If C1, C5 and C7; send the opening RN data queue entry and report message "Missing open RN sent to ARMS".

- When the program reports the first message it encounters based on the condition, the next step is to go to the next ticket.

Process

Hierarchical numeric ID: 1.1.4.3.12.3

Coded name:

Name: RPT Exception of Missing Open Rental Notification

Comment: @Purpose: Synchronize the ECARS and ARMS database by reporting the inconsistency.

@Report Distribution: Not Defined.

ARMS Process Report

Process

Hierarchical numeric ID: 1.1.4.3.13

Coded name:

Name: BAT Online Reporting Sync Report

Comment: @Definition: A batch process that generates a discrepancy report for the ARMS Online Reporting function. This looks at the data stored in file AM095P which is the primary file for the Online Reporting Screens and compares it with ARMS and ARMS/400 detail files to determine if an out of sync condition exists and needs to be addressed.

@Purose: To inform the ARMS application staff of any out of sync conditions that might cause Trading Partner adjusters to see incorrect information on the Online Reporting screens.

@When: This process is run via ROBOT scheduling in the evening.

Process

Hierarchical numeric ID: 1.1.4.3.13.1

Coded name:

Name: PGM CL Wrapper for AMS530A (AML530A)

Comment: @Purpose: This is a CL Wrapper used for the sole purpose of calling AMS530A for creation of the Online Reporting Sync Report.

@Operational Method:

This program does an OVRDBF on file AM097P00 for blocking purposes, adds A4LIB to the library list, OVRPRTF of the report to OUTQ ARMSRPTS, CALLs the program AMS530A and when it returns clears the file AM097P00.

@Files: (CRUD)
-AM097P00 (-R-D)

Process

Hierarchical numeric ID: 1.1.4.3.13.2

Coded name: AMS530A

Name: PGM Online Reporting Sync Report Program (AMS530A)

Comment: @Purpose: This program will look at transactions processed for that day and compare select values between ARMS/400 and ARMS looking for discrepancies. When a difference is found a detail line is written to the report for the use by Oncall personnel to use to be sure that the online report feature stays in sync with ARMS.

@Operational Method:

This program reads through file AM097P00 which is a transactional log created by AM0055V1. This file contains records for both ARMS and ARMS/400 customers and currently the Online Reporting is only used by ARMS/400 customers. Therefore a check is done by reading A4PRF1 to determine if this is a record for an ARMS/400 customer. IF not skip to the next record. If it is an ARMS/400 customer read the associated file record from AM095P00. If a record is found proceed with a sync check. Read the required detail files and those listed in the @Files section.

The following data elements are sync checked:

- Customer Tran. ID
- Branch Claims Office ID
- Claim Number
- Claim Type
- Rental Status
- Adjuster last and first name

ARMS Process Report

Body S name
Rental start date
Rental close date
Rental termination date
Daily authorized rate
Number of days Authorized
Dollars covered per Day
Policy Maximum Dollars
Bill to %
verify correctness of the Surcharge flag
IF Invoiced:
 Total Charges
 Total Amount Received
 Billed Amount

Calculated filelds checked against AM095P00
 Number of days in Rental
 Number of days Behind
 Number of Extensions received

If any of the above mentioned fields are in an out of sync condition
write a detail record to the Sync Report. Continue with the above process until the
file AM097P00 is at End of File.

@Notes: Currently the program only checks transactions that are at Open
Rental status and is not processing the code for invoiced transactions.

@Files: (CRUD)

- A4XREFL7 (-R--)
- A4AUTD (-R--)
- A4RPRD (-R--)
- AMFDTBL (-R--)
- AM095P00 (-R--)
- AM097P00 (-R--)
- A4SURD (-R--)
- A4PRF1 (-R--)
- A4IEBT (-R--)
- AMTIMEL1 (-R--)
- AMTRNLOG (-R--)

@Embedded Data/Constants

Surchg1 = 'Flag is Yes'
Surchg2 = 'No A4SURD rec. found'
Surchg3 = 'Surchg flag'
RntDys1 = 'Days in Rental'
RntDys2 = 'Value in AM095P is not '
RntDys3 = 'correct please verify'
Dysbhnd1 = 'Rental Days Behind'
Dysbhnd2 = 'Value in AM095P is not '
Dysbhnd3 = 'correct please verify'
AdjName = 'Adjuster Name'
NoXref = 'No A4XREF for this Tran'

Process

Hierarchical numeric ID: 1.1.4.3.13.3
Coded name:

ARMS Process Report

Name: RPT Online Sync Report
Comment:

Process

Hierarchical numeric ID: 1.1.4.3.14

Coded name:

Name: BAT Nightly Update Open Tickets for AM095P

Comment: @Definition: A batch process that will update the ARMS Online Reporting database to keep select data elements such as number of days in rental current.

@Purpose: There exists several calculated fields in the database which would become incorrect if a nightly update did not occur. This process maintains the validity and integrity of this data. The file updated is AM095P.

@When: This process is run via ROBOT scheduling in the evening.

Process

Hierarchical numeric ID: 1.1.4.3.14.1

Coded name: AML520A

Name: PGM Nightly Update/Write Open Tickets (AML520A)

Comment: @Purpose: To update all open tickets in the ARMS Online Reporting database file to show the correct

number of days in rental and the number of days behind if the extension date has been passed. This will ensure that open rentals that do not have a transaction processed on a given day will be kept up to date.

@Operational Method:

- Retrieve a record from ARMS Online Reporting Detail File AM095P.

- Determine the total number of days in the rental by calculating the difference between today's date and the rental start date.

- IF the termination date is greater than zero, zero out the days behind. Otherwise, the days behind will be equal to the total number of days in the rental less the number of authorized days.

- Update ARMS Online Reporting Detail File AM095P.

@Files: (CRUD)

- AM0095P01 (-RU-)

@Embedded Data/Constants:

Process

Hierarchical numeric ID: 1.1.4.4

Coded name:

Name: SA Tools

ARMS Process Report

Comment: @Definition: The activity that encompass the manual processes that are performed by various Application Support Staff. These processes range from keeping the environment up and running, troubleshooting transaction problems, to requesting reports.

Process

Hierarchical numeric ID: 1.1.4.4.1

Coded name:

Name: ONL Generate Application Reports

Comment: @Definition:

This activity is performed by ARMS Help Desk and ARMS Technical for generating system utilization by Trading Partner and daily billing error report.

Process

Hierarchical numeric ID: 1.1.4.4.1.1

Coded name:

Name: MNU ARMS Application Reports

Comment: @Definition: see ONL Generate Application Reports

Process

Hierarchical numeric ID: 1.1.4.4.1.17

Coded name:

Name: ONL ARMS Reservation Detail by Company Utilization Management Reports

Comment: @Definition: This activity provides the Application Support Staff the means to produce a utilization management report for a specific Trading Partner.

Process

Hierarchical numeric ID: 1.1.4.4.1.17.1

Coded name:

Name: ONL Utilization Management Report for Specific Trading Partners BCO (Opt -1)

Comment: @Definition: Generate a report showing ARMS Trading Partners Utilization of the Automated Rental Management System for specific Branch Claims Office.

Process

Hierarchical numeric ID: 1.1.4.4.1.17.1.1

Coded name: AM0135

Name: PGM Print Cross - Reference Utilization Report (AM0135)

Comment: @Purpose:

To print daily utilization report including total reservation, cancelled reservation and rentals, by Trading Partner.

@Operational Method:

- Read entire file printing those records that meet the date range supplied to the program.

ARMS Process Report

Process

Hierarchical numeric ID: 1.1.4.4.1.17.1.2
Coded name: AMS002A
Name: PGM Select Transactions by Trading Partner and BCO (AMS002A)
Comment: @Purpose:

To select transactions from the cross-reference file, to report for a specified trading partner or branch claims office.

@Operational Method:

- Select those records that meet the date and status criteria.

Process

Hierarchical numeric ID: 1.1.4.4.1.17.1.4
Coded name: CLL003
Name: PGM Gather / Print Cross - Reference Statistical Report (CLL003)
Comment: @Purpose:

To generate a statistical transactions report using the cross-reference file.

@Operational Method:

- Execute the process that selects those records that meet the criteria for the utilization report, program AMS002A
- Override printer file to outq ARMSRPTS.
- Execute the process that print the utilization report, program AM0135.

Process

Hierarchical numeric ID: 1.1.4.4.1.17.1.5
Coded name: AM0136
Name: PGM Display / Select Available Trading Partners Claims Office (AM0136)
Comment: @Purpose:

To display a list of available claims offices from which the Application Support Staff can select from.

@Operational Method:

- Build a file / list of available Trading Partners.
- When requested, display the filter allowing the user to select specific Trading Partners Branch Claims Office.

Process

Hierarchical numeric ID: 1.1.4.4.1.17.1.6
Coded name: CLL001
Name: PGM Gather / Print ARMS Reservation Detail Utilization Management Reports (CLL001)

ARMS Process Report

Comment: @Purpose:

To allow the "ARMS Technical" staff to monitor trading partner's usage of the ARMS system via a cross-reference report of authorizations approved through ARMS. The filter criterion can specify the trading partner profile (all BCO's, Branch Claims Offices) or particular BCO's within a profile

@Operational Method:

- Print a detailed report by Rental Management Trading Partner company of the current statistics of approved authorizations via ARMS.

@Files: (CRUD)

- AMXREF (-R--)
- ARMSPR2 (-R--)
- AMCLSTBL (-R--)
- AMAUTD (-R--)
- OFFDGRPPF (-R--)
- AMMNTLOG (-R--)

@Note: Used by ARMS Technical Manager to see increase or decrease in ARMS usage volume by each Rental Management Trading Partner.

Process

Hierarchical numeric ID: 1.1.4.4.1.17.2

Coded name:

Name: ONL Utilization Management Report for Trading Partner (Opt - 2)

Comment: @Definition: Generate a report showing ARMS Trading Partners Utilization of the Automated Rental Management System.

Process

Hierarchical numeric ID: 1.1.4.4.1.17.2.1

Coded name: AMS003A

Name: PGM Select Transactions for a Trading Partner (AMS003A)

Comment: @Purpose:

To select transactions from the cross-reference file based on certain criteria specified via a job file AML136JA.

@Operational Method:

- Select those records that meet the date and status criteria.

Process

Hierarchical numeric ID: 1.1.4.4.1.17.2.3

Coded name: CLL001A

Name: PGM Gather / Print Cross - Reference Statistical Report (CLL001A)

Comment: @Purpose:

To set the environment and execute the programs that select transactions from the cross-reference file, and print the actual report.

@Operational Method:

- Execute the program to select the transactions specified by the user.
- Override the print file for the report generator
- Execute the program to print the utilization report.

Process

Hierarchical numeric ID: 1.1.4.4.1.18

Coded name:

Name: ONL Generate Report of Errors Sent by a Trading Partner on an Electronic Invoice

Comment: @Purpose: A process where the Application Support Staff can generate a list of all the sent electronic invoices that were rejected by the trading partner.

Process

Hierarchical numeric ID: 1.1.4.4.1.18.1

Coded name: CLL015

Name: PGM Generate Daily Billing Error Report (CLL015)

Comment: @Purpose:

To control the generation of the daily billing error report.

@Operational Method:

- Execute the program to build the work file with the billing error

Information

- Execute the program to print the billing error report.

Process

Hierarchical numeric ID: 1.1.4.4.1.18.2

Coded name:

Name: PGM ARMS System Control (AAMAIN)

Comment: @Definition: A single source driver for the entire ARMS System Control 'menu'. Allowing the Application Support Staff to maintain the ARMS environment; Trading Partner profiles; application tables;

Process

Hierarchical numeric ID: 1.1.4.4.1.18.6

Coded name: AM5010

Name: PGM Construct Billing Error File (AM5010)

Comment: @Purpose:

To build a work file, selecting records according to the specified date range of error records received from the Trading Partners for sent invoices.

@Operational Method:

- Read entire received bill error file selecting those records that match the date range criteria.
- When a match is found retrieve the associated sent transaction.

Process

Hierarchical numeric ID: 1.1.4.4.1.18.7
 Coded name: AM5015
 Name: PGM Print Billing Error report (AM5015)
 Comment: @Purpose:

To print a list of errors including comments received from the Trading Partner that fall in the specified date range.

@Operational Method:

- Process the work file, print the transaction that rejected and why.

Process

Hierarchical numeric ID: 1.1.4.4.2
 Coded name:
 Name: SCR Work with Robot Scheduler (RBM)
 Comment: @Definition:

This activity routes the ARMS on-call staff to the job scheduler main menu where inquiry about specific jobs can be made.

Process

Hierarchical numeric ID: 1.1.4.4.3
 Coded name:
 Name: ONL Work with Transaction Exception
 Comment: @Definition:

This activity is performed by the ARMS on-call staff to be both reactive and proactive to data base discrepancies between the centralized ARMS system and distributed system as well as ARMS system and Rental system and ARMS/400.

Process

Hierarchical numeric ID: 1.1.4.4.3.1
 Coded name:
 Name: SCR Work with Job (LAMZ001A)
 Comment: @Definition: Provides the ARMS on-call staff an option for retrieving the missing rental notification report.

@Notes:

- This report is presently the proactive approach to the missing rental notification problem. The problem is the result of deficiencies in handling certain scenarios;

- Report is generated on the AEFIKLMNPSTWY host computer platforms.

Process

Hierarchical numeric ID: 1.1.4.4.3.2
 Coded name:
 Name: SCR Work with Job (AML900)

ARMS Process Report

Comment: @Definition: Provides the ARMS on-call staff an option for retrieving the discrepancy report between ARMS and Rental.

@Notes:

- This program runs a series of queries that compares the data found in the Ticket file to ARMS cross-reference, as well as the data found in the reservation file to ARMS cross-reference.

- The report based on the reservation file contains many of transaction that should not be include. This is the result of using the open ticket date on the reservation file instead of the contract date from the ticket.

- The report does show when two reservation points to only one ticket.

- Report is generated on the AEFIKLMNPSTWY host computer platforms.

Process

Hierarchical numeric ID: 1.1.4.4.3.3

Coded name:

Name: SCR Work with Job (L749)

Comment: @Definition: Provides the on-call staff an option for retrieving the ARMS cross-reference discrepancy report.

@Notes:

- Report is generated on RARMS machine.

Process

Hierarchical numeric ID: 1.1.4.4.3.4

Coded name:

Name: SCR Search for Spool Entries (DSPOLE)

Comment: @Definition: Provides the ARMS on-call staff an option for retrieving the electronic billing error report that has been archived

@Notes:

- Report is generated on AEFIKLMNPSTWY host computer platforms.

Process

Hierarchical numeric ID: 1.1.4.4.3.5

Coded name:

Name: SCR Work with All Spooled Files (A4ERROR)

Comment: @Definition: Provides the ARMS on-call staff an option for retrieving the discrepancy report between ARMS and ARMS/400.

@Notes:

- Report runs on RARMS only.

- Libraries A4LIB and ELPGMR required.

Process

Hierarchical numeric ID: 1.1.4.4.3.6

Coded name:

Name: SCR Work with All Spooled Files (A4XREF)

ARMS Process Report

Comment: @Definition: Provides the ARMS on-call staff an option for retrieving the report of ARMS/400 transactions at a sent status.

@Notes:

- Reactive approach for handling errors.
- Report runs on RARMS only.
- Libraries A4LIB and ELPGMR required

Process

Hierarchical numeric ID: 1.1.4.4.3.7

Coded name:

Name: MNU ONCALL Reports

Comment: @Definition: see ONL Work with Transaction Exception

Process

Hierarchical numeric ID: 1.1.4.4.4

Coded name:

Name: SCR Access ARMS/400 Software (A4000RMS)

Comment: @Definition: This activity allows the ARMS on-call staff to access the ARMS/400 software for researching problems.

@Notes: This access path to the ARMS/400 interactive system allows a user to see the screens and information in the same way that the adjuster or in-house Enterprise employee would. Further this was created initially for use by the Rental Management Services team headed by Jim Hillerman.

Process

Hierarchical numeric ID: 1.1.4.4.5

Coded name:

Name: ONL Perform ARMS Environment Maintenance

Comment: @Definition:

This activity is performed by the ARMS on-call staff for managing the application environment (ARMS subsystem).

Process

Hierarchical numeric ID: 1.1.4.4.5.1

Coded name:

Name: ONL Start Individual Trading Partner Communication Programs

Comment: @Definition:

This activity is performed by the staff on call to select and re-start the communication programs for a given Trading Partner when they have abnormally ended.

@Notes:

- Runs on RARMS

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.4.4.5.1.1
Coded name: CCARMS
Name: PGM Select Company / Machine (CCARMS)
Comment: @Purpose:

To allow to select and return the selected Trading Partner send and/or receive detail to provide a way to re-start the communication programs for a given Trading Partner when they have abnormally ended.

@Operational Method:

- Display the list of computer platforms where ARMS can be started.
- For each machine selected, read the ARMS profile attributes for that trading partner and display the send and receive and associated translator and mapper program START FLAGS for the user to select which programs to start.
- Return the flag values as selected by the user.

Process

Hierarchical numeric ID: 1.1.4.4.5.1.2
Coded name: AMSTARTR
Name: PGM Start Specific Trading Partner Send & Receive pgms (AMSTARTR)
Comment: @Purpose:

To submit jobs to start the send and receive and associated programs for each Specified Trading Partner without a VAN in the profile file (ARMSPR3).

@Operational Method:

- If the company id is passed,
 retrieve the communication program technical detail
 for the passed company id
 Else all companies programs to be started
 read the file ARMSPR3 (ARMS Technical Profile) sequentially
 and perform the following steps for each qualified company
- Based on the profile, the passed flags and some hard coded logic,
submit jobs to start appropriate communication programs at appropriate scheduled times.

Process

Hierarchical numeric ID: 1.1.4.4.5.2
Coded name:
Name: ONL Start Transaction Distribution Programs for Specific(s) Machine
Comment: @Definition: A tool which provides the Application Support Staff the means necessary to start the distribution program from the centralized host machine to the distributed host machine.

@Notes:

- Runs on RARMS

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.4.4.5.2.2
Coded name: DQAAARMS
Name: DTQ Selected Computer platforms with Start Flags
Comment: @Definition:

This data queue is created by program AAMAIN in library QTEMP to retrieve the selected computer platforms and associated start flags. The computer platforms are selected in program CCARMS where the entries are sent to this data queue.

Essentially this data queue acts as a data area to communicate between the two programs AAMAIN and CCARMS.

Process

Hierarchical numeric ID: 1.1.4.4.5.3
Coded name:
Name: ONL Start the ARMS to Rental Systems Interfaces
Comment: @Definition: A tool which provides the Application Support Staff the means necessary to start all of the ARMS to Rental Systems Interfaces for a single distributed host machine.

@Note:
- Runs on A,C,E,F,I,K,L,M,N,P,S,T,W,Y

Process

Hierarchical numeric ID: 1.1.4.4.5.3.1
Coded name: AMSTARTE
Name: PGM Start the ARMS - Rental System Interfaces (AMSTARTE)
Comment: @Purpose:

To allow the user to start the rental system interface programs for a particular machine.

@Operational Method:

- Submit jobs to start each of the ECARS interface programs.

@Notes: Interface include: EC/CC00ATVn, EC/CC00CNVn, EC/CC00EXVn, EC/CC00ACVn, EC/CC00RAVn, EC/CC00RCVn, EC/CC00REVn, EC/CC00RNVn, EC/CC00TRVn, EC00CMVn, EC00VMVn, CC00INVn, AM0061Vn, AM0060Vn

Process

Hierarchical numeric ID: 1.1.4.4.5.4
Coded name: WRKCFGSTS
Name: ONL Work with a Specific Trading Partner Communication Line (WRKCFGSTS)
Comment: @Definition: A command that allows the Application Support Staff to display and work with the status of a Specific Trading Partner's communication line.

@Notes:
- Run on RARMS

Process

Hierarchical numeric ID: 1.1.4.4.5.5

ARMS Process Report

Coded name: WRKLIND

Name: ONL Work with a Specific Trading Partner Communication Line Description (WRKLIND)

Comment: @Definition: A command that allows the Application Support Staff to display and work with the line description of a Specific Trading Partner.

@Notes:

- Run on RARMS

Process

Hierarchical numeric ID: 1.1.4.4.5.6

Coded name: WRKHDWRSC

Name: ONL Work with Local Communication Resources (WRKHDWRSC)

Comment: @Definition: A command that shows information about local communications resources on the system and the associated configuration objects, including all communication IOPs, IOAs, and ports installed on the system.

@Notes:

- Run on RARMS

Process

Hierarchical numeric ID: 1.1.4.4.5.7

Coded name:

Name: ONL Work with the ARMS Subsystem (WRKSBSJOB)

Comment: @Definition: A command that shows all jobs running in the Application Specific subsystem (ARMS).

@Note: Subsystem is the same on all computer platforms. Jobs running in the subsystem differs from centralized host computer platforms to distributed host computer platforms to the Financial Application systems host computer.

- Runs everywhere

Process

Hierarchical numeric ID: 1.1.4.4.5.8

Coded name: NEWARMS

Name: ONL Start new Subsystem Program (NEWARMS)

Comment: @Purpose: A command that allows the Application Support Staff to start a user specified, non-data required program.

@Notes:

- Runs everywhere

Process

Hierarchical numeric ID: 1.1.4.4.5.9

Coded name:

Name: ONL Shutdown ARMS Environment for a Specific Computer platforms

Comment: @Purpose: The process by which the ARMS Application Support Staff ends the processes running on a specific distributed computer platforms.

@Notes: This usually is the result of a computer platforms that is experiencing hardware difficulties and OPS is wants to bring it down.

- Run on RARMS only

Process

Hierarchical numeric ID: 1.1.4.4.5.9.1

Coded name: AMSHUTDOWN1

Name: PGM Send Shutdown Request (AMSHUTDOWN1)

Comment: @Purpose: The process where a shutdown request is sent by the ARMS on-call staff, for a specific distributed computer platforms, to end the running ARMS environment.

@Operational Method:

- If running on the RARMS computer platforms, send data entry to DQAMAP1 with the specified computer platforms prefix as the key.

@Notes:

- *DOWN00000000000000SD*DOWN1 = Shutdown data queue entry.

Process

Hierarchical numeric ID: 1.1.4.4.5.10

Coded name:

Name: ONL Display the Number of Shutdown the Network Requires

Comment: @Definition: Process by which the ARMS Application Support Staff can see the number of shutdown records the ARMS network requires before shutting down.

@Notes:

- The number of shutdown records is derived from the number of distributed computer platforms it is running on. Presently, the number is 14.

- DSPDTAARA AM001A

- Runs on RARMS only

Process

Hierarchical numeric ID: 1.1.4.4.5.11

Coded name:

Name: ONL Display the Number of Shutdown Network Received

Comment: @Definition: Process by which the ARMS Application Support Staff can see the number of shutdown records the ARMS network has received.

@Notes:

- If not enough shutdown records have been received the on-call staff can generate the missing records to cause the ARMS network to shutdown.

**The most common cause for insufficient shutdown records happens when one of the distributed computer platforms is brought down at an unscheduled time.

- DSPDTAARA AM002A

- Runs on RARMS

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.4.4.5.12

Coded name:

Name: ONL Work with the ARMS Job Queue (WRKJOBQ)

Comment: @Definition: Process by which the ARMS Application Support Staff can monitor the ARMS job queue.

@Notes:

- Run everywhere

Process

Hierarchical numeric ID: 1.1.4.4.5.13

Coded name:

Name: ONL Display the Number of any ARMS-connected VAN Receivers Started

Comment: @Definition: Process by which the ARMS Application Support Staff can see the number of ARMS-connected VAN communication receiver jobs started.

@Notes:

- This count is usually 1. However, if a second receiver is started the data queue and file that the received transmissions are placed into is AMVN101T instead of AMVN1011.

- DSPDTAARA AM003A
- Run on RARMS

Process

Hierarchical numeric ID: 1.1.4.4.5.14

Coded name:

Name: ONL Display the Number of Transaction Translators Running

@Definition: The process by which the ARMS on-call staff can see the number of translators running.

@Notes:

- This is important when shutting down the ARMS network normally. If the value is not equal to zero then a translator is active and the ARMS network will NOT shutdown.

** If the translator was ended abnormally the count is not adjusted.

- DSPDTAARA DASDV1
- Run on RARMS

Process

Hierarchical numeric ID: 1.1.4.4.5.15

Coded name:

Name: MNU Work with ARMS Environment

Comment: @Definition: see ONL Perform ARMS Environment Maintenance

Process

Hierarchical numeric ID: 1.1.4.4.6

Coded name:

Name: ONL Perform ARMS/400 Environment Maintenance

ARMS Process Report

Comment: @Definition

This activity is performed by the ARMS on-call staff for managing the ARMS/400 application environment.

@Notes:

- Runs on RARMS only

Process

Hierarchical numeric ID: 1.1.4.4.6.1

Coded name:

Name: ONL Work with the ARMS/400 Subsystem (WRKSBSJOB)

Comment: @Definition: A command that shows all jobs running in the Application Specific subsystem (A4BAT).

@Note: Subsystem only on RARMS

Process

Hierarchical numeric ID: 1.1.4.4.6.2

Coded name:

Name: ONL Work with the ARMS/400 Job Queue (WRKJOBQ)

Comment: @Definition: Process by which the ARMS Application Support Staff can monitor the A4BAT job queue.

@Notes:

- Run RARMS only

Process

Hierarchical numeric ID: 1.1.4.4.6.3

Coded name: NEWARMS400

Name: ONL Start new Subsystem Program (NEWARMS400)

Comment: @Purpose: A command that allows the Application Support Staff to start a user specified, non-data required program.

@Notes:

- Runs RARMS only

Process

Hierarchical numeric ID: 1.1.4.4.6.4

Coded name:

Name: MNU Work with ARMS/400 Environment

Comment: @Definition: see ONL Perform ARMS/400 Environment Maintenance

Process

Hierarchical numeric ID: 1.1.4.4.7

Coded name: AAMAIN

Name: MNU ARMS Main Menu (AAMAIN)

Comment: @Definition: A consolidated menu where by the ARMS on-call staff can find and access support tools needed to perform their daily activities.

ARMS Process Report

@Notes:

- Rental System Help Desk and ARMS Application Support Service use the menu.
- Tools are group into four main categories, with the further group into subcategories and so on based on their related functions:
 - Reports
 - Application
 - On-call
 - Applications
 - Robot Scheduler
 - ARMS/400
 - Maintenance / Inquiry
 - ARMS Environment
 - Startup Controls
 - Shutdown Controls
 - Communication Status
 - Other
 - ARMS/400 Environment
 - Transaction Log
 - Data Queue
 - Reservation/Ticket Information
 - Purge Progress
 - File
- Other

Process

Hierarchical numeric ID: 1.1.4.4.8

Coded name:

Name: ONL Inquire on Transaction Log

Comment: @Definition:

This activity is used by on-call and help desk when troubleshooting discrepancies between the Trading Partner and Rental Systems

Process

Hierarchical numeric ID: 1.1.4.4.8.1

Coded name:

Name: MNU ARMS System Control Transaction Log

Comment: @Definition: see ONL Inquire on Transaction Log

Process

Hierarchical numeric ID: 1.1.4.4.8.11

Coded name:

Name: ONL Inquiry of Transactions Received from a Trading Partner

Comment: @Definition: The process which allows the Application Support Staff to select specific transactions received from the Trading Partner's for inquiry.

Process

Hierarchical numeric ID: 1.1.4.4.8.11.1

Coded name: CCAM12

Name: PGM Transaction Receive Log Inquiry (CCAM12)

ARMS Process Report

Comment: @Purpose:

To allow the Application Support Staff to select specific transactions received from the Trading Partners for inquiry.

@Operational Method:

- Display a list of all the transmission received from specific Trading Partners.
- Upon the select of an individual transmission, display the control/data records the make up that transmission.

Process

Hierarchical numeric ID: 1.1.4.4.8.12

Coded name:

Name: ONL Inquiry of Error Transactions Sent by Trading Partner

Comment: @Definition: A process where the Application Support Staff has the ability to view error transactions sent by Trading Partners.

Process

Hierarchical numeric ID: 1.1.4.4.8.12.1

Coded name: CCAM13

Name: PGM Receive / Billing Error Inquiry (CCAM13)

Comment: @Purpose:

To display either the billing errors received from the Trading Partners or the non - billing errors. This is helpful for troubleshooting.

@Operational Method:

- Based on the parameter passed to the program open either the received error log file or the billing error log file.
- Read all records from the file, displaying the records on screen.
- Upon the selection of an individual transmission, display the error code received.

Process

Hierarchical numeric ID: 1.1.4.4.8.13

Coded name:

Name: ONL Inquiry of Transaction Sent by the Trading Partner but Erred / Rejected

Comment: @Definition: The process where the Application Support Staff can display the errors that have been sent to the Trading Partners.

Process

Hierarchical numeric ID: 1.1.4.4.8.13.1

Coded name: CCAM14

Name: PGM Error Log Inquiry (CCAM14)

ARMS Process Report

Comment: @Definition: The process where the Application Support Staff can display the errors that have been sent to the Trading Partners.

@Purpose: Used in trouble shooting problems

@Operational Method:

- Display a list of the errors received by Enterprise from a specific Trading Partner.

- When a specific transmission is selected, display the data records associated with that transmission.

Process

Hierarchical numeric ID: 1.1.4.4.8.14

Coded name:

Name: ONL Inquiry / Resend of Transaction(s) ARMS Sent to Trading Partner

Comment: @Definition: A process where by the Application Support Staff can display transaction sent to the Trading Partner.

Process

Hierarchical numeric ID: 1.1.4.4.8.14.1

Coded name: CCAM15

Name: PGM Send Log Inquiry and Resend (CCAM15)

Comment: @Definition: A program where by the Application Support Staff can display transaction sent to the Trading Partner. It may, if needed, select a sent transaction, alter the data, and resend the transaction to the Trading Partner for processing.

@Purpose: This is useful when troubleshooting a problem

@Operational Method:

- Read the sent transaction log file, display a list on screen for the user.

- Upon the selection of one of the sent transmissions, display the data/control record that make up that single transmission.

- Once a transmission has been selected, the user has the option of changing the data in the transmission and resending it.

Process

Hierarchical numeric ID: 1.1.4.4.8.15

Coded name:

Name: ONL Audit of Transaction Activity

Comment: @Definition: A process where by the Application Support Staff is able to display the electronic history of a specific transaction.

@Purpose: Very useful in problem determination.

Process

ARMS Process Report

Hierarchical numeric ID: 1.1.4.4.8.15.1

Coded name: CCAM16

Name: PGM Timeline Inquiry (CCAM16)

Comment: @Definition: A program that will display the electronic history of a specified transaction.

@Purpose: Very useful in problem determination.

@Operational Method:

- Retrieve the transaction information via 1 of 4 ways:
 - Vendor Transaction Id
 - Company & Customer Transaction ID
 - Location & Reservation
 - Location & Ticket
- Retrieve all sent and received transaction with the same vendor transaction id and display this information.
- When the user selects and individual transmission, display the details of that transmission.
- When F2 is pressed, a printout is generated with all the information associated to the transaction.

Process

Hierarchical numeric ID: 1.1.4.4.8.15.2

Coded name: CLL800

Name: PGM Cross-reference Maintenance Log Driver (CLL800)

Comment: @Purpose: The driver program that sets up the print file before executing the program to print the cross-reference maintenance log.

@Operational Method:

- Override print file
- Execute program that generates the maintenance log report

Process

Hierarchical numeric ID: 1.1.4.4.8.15.3

Coded name: AM0800R

Name: PGM Print Database / Cross-Reference Maintenance Log (AM0800R)

Comment: @Purpose: To take a 'snapshot' of the existing database for a single transaction.

@Operational Method:

- Retrieve all associated database records for the single vendor transaction id and print the information.

Process

Hierarchical numeric ID: 1.1.4.4.8.16

Coded name:

Name: ONL Inquiry / Resend of an Electronic Invoice

ARMS Process Report

Comment: @Description: A process where by the Application Support Staff can correct an electronic invoices and send it to the Trading Partner.

Process

Hierarchical numeric ID: 1.1.4.4.8.16.1

Coded name: AAAM15

Name: PGM Billing Log Inquiry and Resend (AAAM15)

Comment: @Purpose: To select/display records for billing corrections and to RESEND to the customer.

@Operational Method:

- The user must first select the order in which the data will be displayed (GPBR, TICKET# or LOCATION, TICKET#).
- Then the user will select the item to be changed and make the change. The program then recalculates the totals for the trailer record and redisplay all. The user can then choose to resend this billing to the customer by pressing CMD-9 twice.

- This will write the header, all corrected detail, and the trailer records to AMPACK. Also will write the APPD01 record before the header record.

Process

Hierarchical numeric ID: 1.1.4.4.9

Coded name:

Name: ONL Perform Data Queue Maintenance

Comment: @Definition:

This activity is performed by the ARMS on-call staff to support the application by manipulating the data (transactions) in the environment.

Process

Hierarchical numeric ID: 1.1.4.4.9.1

Coded name: ARMSONCL

Name: SCR ARMS Data Queue Monitor (ARMSONCL)

Comment: @Definition: A process from which the ARMS on-call staff can monitor the number of messages waiting to process and where.

@Notes:

- Run everywhere

Process

Hierarchical numeric ID: 1.1.4.4.9.2

Coded name: A4ONCL

Name: SCR ARMS/400 Data Queue Monitor (A4ONCL)

Comment: @Definition: A process from which the ARMS on-call staff can monitor the number of messages waiting to process and where.

@Notes:

- Run RARMS only

ARMS Process Report

Process

Hierarchical numeric ID: 1.1.4.4.9.3

Coded name: ANDQLDDQ

Name: ONL Load Data Queue with Message from Data Queue (ANDQLDDQ)

Comment: @Definition: A process where the ARMS on-call staff can read and load data queue message from and to a specified data queue.

@Purpose: Provide the application development staff a way to perform load balancing when problems with external communications has resulting if a backlog of un-sent transactions.

@Note:

- User specifies the number of messages to read, starting at the first entry.

Process

Hierarchical numeric ID: 1.1.4.4.9.4

Coded name: ANDQLDPF

Name: ONL Load Data Queue with Message from a Physical File (ANDQLDPF)

Comment: @Definition: A process where the ARMS on-call staff can read record from a specified physical file and load data queue message to a specified data queue.

@Purpose: Provide the application development staff a way to generate the corresponding data queue entry for the data records.

@Notes:

Process

Hierarchical numeric ID: 1.1.4.4.9.5

Coded name:

Name: ONL Retrieve Data Queue Information

Comment: @Definition: A process that allows the ARMS on-call the option to retrieve information about the data queue, particularly the number of messages.

Process

Hierarchical numeric ID: 1.1.4.4.9.6

Coded name: RGZDTAQ

Name: ONL Reorganize Data Queue (RGZDTAQ)

Comment: @Definition: The process where by the ARMS on-call staff can delete and recreate a data queue without destroying the data queue entries that might still exist.

@Purpose: Data queue processing some known traits - If the number of messages grow to more than 100, the size of the data queue expands. The only way to reclaim the space is to delete and recreate the data queue. Its a performance issue.

Process

Hierarchical numeric ID: 1.1.4.4.9.7

Coded name:

Name: MNU ARMS System Data Queue Maintenance

ARMS Process Report

Comment: @Definition: See ONL Perform Data Queue Maintenance

Process

Hierarchical numeric ID: 1.1.4.4.9.8

Coded name:

Name: ONL Work with Application Data Queues

Comment: @Definition: A process that provides the Application Support Staff easy access to manage / manipulate the data being passed between processes.

@Purpose: Provide easy access of the APIs to display, dump, and clear a data queue as well as add a data queue entry through the use of a display screen.

Process

Hierarchical numeric ID: 1.1.4.4.9.8.1

Coded name: ANDQMNEU

Name: MNU Application Development Data Queue Menu

Comment: @Definition: A menu that provides the Application Support Staff easy access to manage/manipulate the data being passed between processes.

@Purpose: Provide easy access of the APIs to display, dump, and clear a data queue as well as add a data queue entry through the use of a display screen.

Process

Hierarchical numeric ID: 1.1.4.4.9.8.3

Coded name:

Name: SCR Add a Data Queue Entry

Comment: @Definition: A tool that allows the Application Support Staff to enter data queue entries as needed.

Process

Hierarchical numeric ID: 1.1.4.4.10

Coded name:

Name: ONL Perform Application File Maintenance

Comment: @Definition:

This activity is performed by ARMS Technical to maintain various application table files (i.e. BCO, Trading Partner Profile)

Process

Hierarchical numeric ID: 1.1.4.4.10.1

Coded name:

Name: ONL Maintain Trading Partner Profile Global Attributes

Comment: @Definition: The ARMS Technical Staff uses this activity to maintain the ARMS customer global profile information like how the customer connects to Enterprise, what types of transactions the customer receives and/or sends, what is the time window for the customer during which batch invoices can be sent etc.

ARMS Process Report

Process

Hierarchical numeric ID: 1.1.4.4.10.1.1
Coded name: AM0700
Name: PGM Maintain ARMS Trading Partner Profiles (AM0700)
Comment: @Purpose:

To allow the user to maintain all the company profile attributes related to transactions and communications.

@Operational Method:

This program displays a list of all the companies that have been set up in the ARMS system and has options for the user to add, change, copy, delete or display profiles.

There is a separate function key for each classification of the profile attributes and a separate program maintains each classification. The following functions are available:

F7=Appl Specific
F8=Name Addr
F9=Tech Data
F10=File/Field Ctrl
F14=ECARS Data
F16=Claims
F17=Discount
F18=Underage Driver

Process

Hierarchical numeric ID: 1.1.4.4.10.1.2
Coded name: AM0701
Name: PGM Maintain ARMS Trading Partner Profile Application Specific Information (AM0701)
Comment: @Purpose:

To maintain the ARMS Application Specific Data that defines how the trading partner host system interacts with the ARMS system. This file is essentially contains an electronic agreement between the two.

@Operational Method:

- This program gets invoked when the user presses function key F7 to maintain the application specific data from the main screen displayed by controlling program AM0700.

- The user can change all the profilable attributes for any company. The attributes and their description are retrieved from the ARMS field table file AMFDTBL.

@Files:

ARMSPR1 (-RU-)
ARMSPR2 (-R--)
AMFDTBL (-R--)

@Notes:

ARMS Process Report

The changes are then made online to the distributed files via DDM links.

Process

Hierarchical numeric ID: 1.1.4.4.10.1.3

Coded name: AM0702

Name: PGM Maintain ARMS Trading Partner Profile Contact Information (AM0702)

Comment: @Purpose:

To maintain the ARMS Customer Address Data for the insurance company.

@Operational Method:

- This program gets invoked when the user presses function key F8 to maintain the company name and address information from the controlling program AM0700 main display.

- The user can change the customer's name, address, phone and fax number of the insurance company's corporate office.

- The file ARMSPR2 is keyed by company profile and customer code which allows for more than one record for every profile.

@Notes:

The changes are then made online to the distributed files via DDM links.

Process

Hierarchical numeric ID: 1.1.4.4.10.1.4

Coded name: AM0703

Name: PGM Maintain ARMS Trading Partner Profile Technical Information (AM0703)

Comment: @Purpose:

To maintain the ARMS technical connection information for connecting Enterprise ARMS hostsystem to trading partner host system.

@Operational Method:

- This program gets invoked when the user presses function key F9 to maintain the company technical information from the controlling program AM0700 main display.

- The user can change the customer's connection specific information.

@Notes:

The changes are then made online to the distributed files via DDM links.

Process

Hierarchical numeric ID: 1.1.4.4.10.1.5

Coded name: AM0704

Name: PGM Maintain ARMS Trading Partner Profile File/Field Controls (AM0704)

Comment: @Purpose:

ARMS Process Report

To maintain the insurance company's field control table that determines which fields are mandatory to send and or receive.

@Operational Method:

- This program gets invoked when the user presses function key F10 to maintain the field control information from the controlling program AM0700 main display.

- The user can change the customer's field control table, ARMSPR4.

@Notes:

The changes are then made online to the distributed files via DDM links.

Process

Hierarchical numeric ID: 1.1.4.4.10.1.6

Coded name: AM0705

Name: PGM Maintain ARMS Trading Partner Profile ECARS-Specific Information (AM0705)

Comment: @Purpose:

To maintain the insurance company's ECARS application system specific data like which billing program to call when closing a ticket.

@Operational Method:

- This program gets invoked when the user presses function key F14 to maintain the ECARS information from the controlling program AM0700 main display.

- The user can change how ECARS system interacts with the customer company's authorized rentals.

@Notes:

The changes are then made online to the distributed files via DDM links.

Process

Hierarchical numeric ID: 1.1.4.4.10.1.7

Coded name: AM0706

Name: PGM Maintain ARMS Trading Partner Profile Claims Connection Specific Information (AM0706)

Comment: @Purpose:

To maintain the insurance company's Claims Connection application system specific data like which billing program to call when closing a ticket.

@Operational Method:

- This program gets invoked when the user presses function key F14 to maintain the Claims information from the controlling program AM0700 main display.

- The user can change how Claims Connection system interacts with the customer company's authorized rentals.

@Notes:

ARMS Process Report

The changes are then made online to the distributed files via DDM links.

Process

Hierarchical numeric ID: 1.1.4.4.10.1.8

Coded name: AM0707

Name: PGM Maintain ARMS Trading Partner Profile Volume Discount Levels (AM0707)

Comment: @Purpose:

To maintain the insurance company's discounts based on transaction volumes.

@Operational Method:

- This program gets invoked when the user presses function key F17 to maintain the Discount information from the controlling program AM0700 main display.
- The user can change how much discount a company gets at what threshold levels of transactions.

@Notes:

The changes are then made online to the distributed files via DDM links.

Process

Hierarchical numeric ID: 1.1.4.4.10.1.9

Coded name: AM0708

Name: PGM Maintain ARMS Trading Partner Profile Underage Control by State/Province (AM0708)

Comment: @Purpose:

To maintain the underage surcharge amount and age that is considered underage for each profile and state combination.

@Operational Method:

- This program gets invoked when the user presses function key F18 to maintain the Underage information from the controlling program AM0700 main display.
- The user can change the amount to charge for underage drivers and the legal age for that state by company profile.

@Notes:

- This file is currently empty and no company has been so customized.
- The changes are then made online to the distributed files via DDM links.

Process

Hierarchical numeric ID: 1.1.4.4.10.3

Coded name:

Name: ONL Maintain Application Transaction Error Code Table

Comment: @Definition: A process by which the application support staff can maintain ARMS transaction error code table

Process

Hierarchical numeric ID: 1.1.4.4.10.3.1
Coded name: AAEC01
Name: PGM Maintain Error Code Table (AAEC01)
Comment: @Purpose: Allow the application support staff to maintain the transaction error code table.

Process

Hierarchical numeric ID: 1.1.4.4.10.4
Coded name:
Name: ONL Maintain Car Class Cross-Reference
Comment: @Definition: A process whereby the application support staff maintains the car class (make/model) cross-reference for every Trading Partner.

Process

Hierarchical numeric ID: 1.1.4.4.10.5
Coded name:
Name: ONL Maintain Trading Partner Profile Geographic/Regional Attributes
Comment: @Definition: This activity is used by the ARMS Technical Staff members to maintain the cross-reference between the ARMS customer branch claims office (BCO) and Enterprise customer assigned to the BCO.

Process

Hierarchical numeric ID: 1.1.4.4.10.5.1
Coded name: AAAM10
Name: PGM Maintain Trading Partner Profile Individual Office Locations Attributes (AAAM10)
Comment: @Purpose:
To maintain the cross-reference table between the insurance customer's branch claims offices and Enterprise customer numbers assigned to those branch claims offices.

@Operational Method:

This program reads file AMXBCO and displays the company id and branch claims offices in that order in a subfile display along with the assigned Enterprise customer numbers and claims office description.

The user can position to a particular profile and then choose to maintain details for or delete a particular company id-claims office pair.

When the user selects to maintain, another screen displays the following attributes for the pair:

ARMS Profile ID :
Customer Branch Claims Office . :
Enterprise Customer Number. . . :
BCO Description :

ARMS Process Report

Country Code :
Electronic & Paper Bills. . . . : (Yes or No)
Include Airport Locations.....
ARMS Special Condition :
Default Node ID..... :

ARMS Active Control : Effective start and end dates
ARMS Billing Active Control . . . : - " -
ARMS Payments Active Control. . : - " -

The user has the ability to delete a pair by selecting a 'D' against the pair or another pair can be added by function key F6.

Process

Hierarchical numeric ID: 1.1.4.4.10.6
Coded name:
Name: ONL Maintain Application Master Files/Tables
Comment: @Definition: This activity allows the user to maintain the master files that may contain customized data for a certain company or certain default information to use when generating transactions.

Process

Hierarchical numeric ID: 1.1.4.4.10.6.1
Coded name: AMS004A
Name: PGM Maintain Surcharge Table (AMS004A)
Comment: @Purpose:

To allow the user to set up automatic surcharge proprietary formats (SURD01) by company profile and state. These are surcharges that the company authorizes Enterprise to automatically charge against a rental authorized by that company.

@Operational Method:

The program displays the surcharge format detail fields for a profile in the header section and all the states in the detail section using a standard subfile approach and allows the user to add, change, copy or delete records.

Process

Hierarchical numeric ID: 1.1.4.4.10.7
Coded name:
Name: MNU Maintain Application Files
Comment: @Definition: see ONL Perform Application File Maintenance

Process

Hierarchical numeric ID: 1.1.4.4.10.9
Coded name:
Name: ONL ARMS/400 Training Administration
Comment: @Definition: Process by which the Application Support Staff manage the ARMS/400 Training System.

ARMS Process Report

Process

Hierarchical numeric ID: 1.1.4.4.10.9.1

Coded name: A4TRNGMN

Name: MNU ARMS/400 Training Administration Menu (A4TRNGMN)

Comment: @Definition: A process which enables the Application Support Staff to manage the training environment / data. From setting up claims office and testing scenarios to checking the status of the system.

Process

Hierarchical numeric ID: 1.1.4.4.10.10

Coded name:

Name: ONL Maintain Vehicle Class / Rate Table

Comment: @Definition: Provide the means for the Application Support Staff to maintain Trading Partner's rate to Enterprise's car classes.

Process

Hierarchical numeric ID: 1.1.4.4.10.10.1

Coded name: CCAM11

Name: PGM Vehicle Class / Rate Table Maintenance (CCAM11)

Comment: @Purpose:

To provide the means for the Application Support Staff to maintain Trading Partner's rate to Enterprise's car classes cross-reference table (including ARMS/400 Customers).

@Operational Method:

- Display the details in file AMCLSTBL or A4CLSTBL and allow the user to maintain the data and synchronize the data on all computer platforms via DDM links.

The first screen displays the Company ID, ARMS class, Enterprise class, Class description and the X12 car class.

Options available from this screen:

1. Position to a particular Company ID
2. Add a new ARMS class to an existing Company ID or a new Company ID, in which the Company will be created.
3. Change a current ARMS class for a Company ID.
4. Delete a current ARMS class for a Company ID.
5. Copy an existing ARMS class to a new or existing Company ID.

When the user selects to add or change an ARMS class another screen is displayed. This screen will allow the user to change or input an ARMS class, Enterprise class, class description and X12 car class.

When the user chooses to copy one or more ARMS classes, a screen will be displayed with the chosen ARMS classes, Enterprise classes, class descriptions and X12 car class, from which the user can choose a company ID, new or preexisting, to copy all of the classes to.

@Files:

AMCLSTBL (C__)

A4CLSTBL (C__)

DCSPCL (R__)

DCSPTY (R__)

ARMS Process Report

DCSPSH ()
DCSPAC (R)
ARMSPR1 (R)
ARMSPR3 (R)

Process

Hierarchical numeric ID: 1.1.4.4.11

Coded name: AM0900

Name: SCR Create Invoice Request (AM0900)

Comment: @Definition:

This activity is performed by the help desk to generate paper and electronic bills for previously errored invoices.

@Operational Method:

The user input fields include group, branch, and ticket number. AM0900 first verifies that the ticket is in the file, QRACCLSQ (closed ticket file). Secondly, it verifies that it is an ARMS ticket by checking a field in RACSLD. Finally AM0900 calls AM1010V1 that retrieves the corresponding AMXREF record.

Based on the status code returned, the program allows the user to proceed in AM0900. The next screen allows the user to hit a function key for re-billing and selecting of either paper or electronic billing method. A record is then written to RACBILL to reflect the rebill choice and an AMBILL record id written to log the rebill.

@Notes:

- Runs on A,E,F,I,K,L,M,N,P,S,T,W,Y

@Files: (CRUD)

-QRACCLSQ (C) (Logical over RACCLSC and past 12 monthly files)

-RACSLD (R) (Logical over RACSLC and past 12 monthly files)

-RACBILL (C)

-AMBILL (C)

Process

Hierarchical numeric ID: 1.1.4.4.12

Coded name: ARMSEDT

Name: SCR Display Reservation / Ticket Information (ARMSEDT)

Comment: @Definition: A process that provides the ARMS on-call staff the option of retrieving specific data elements from multiple Rental files.

@Notes:

Process

Hierarchical numeric ID: 1.1.4.4.13

Coded name:

Name: ONL Monitor Purge Activity

Comment: @Definition:

This activity is used by ARMS on-call to monitor the progress of the ARMS application database purge.

ARMS Process Report

Process

Hierarchical numeric ID: 1.1.4.4.13.1

Coded name:

Name: ONL Display Database Transaction Set Controls

Comment: @Definition: A process where by the ARMS on-call staff can monitor the database purge.

@Notes:

- DSPDDTAARA AM006A
- Three steps involved in the entire purge process.

Process

Hierarchical numeric ID: 1.1.4.4.13.2

Coded name:

Name: ONL Display Orphaned Transaction Set Controls

Comment: @Definition: A process where by the ARMS on-call staff can monitor the database purge.

@Notes:

- DSPDDTAARA AM009A
- Three steps involved in the entire purge process.

Process

Hierarchical numeric ID: 1.1.4.4.13.3

Coded name:

Name: ONL Display Reorganize File Control

Comment: @Definition: A process where by the ARMS on-call staff can monitor the database purge.

@Notes:

- DSPDDTAARA AM009A
- Three steps involved in the entire purge process.

Process

Hierarchical numeric ID: 1.1.4.4.13.4

Coded name:

Name: MNU Display Purge Programs

Comment: @Definition: see ONL Monitor Purge Activity.

Process

Hierarchical numeric ID: 0

Coded name:

Name: Rental and ARMS

Comment: @Definition: This is the Automated Rental Management Systems (ARMS) application context as it currently is.